

浙江大学



VAG 研究生新生指南

第一版:

作者: 陈为 丁子昂 严志程 于洋 叶棣

修订: 张涛

第二版:

修订: 陈伟锋



E-mail: vag.zjucadcg@gmail.com

修订日期: April 8, 2012

目 录

目录

1	VAG 小组日常制度	1
1.1	周报制度	1
1.2	出勤制度	2
1.3	学术讨论制度	2
1.4	经费报销制度	4
1.4.1	经费来源	4
1.4.2	报销制度	5
1.4.3	公共经费	6
1.4.4	发票报销	6
1.5	科研工作展示	8
1.5.1	论文准备	8
1.5.2	论文发表	8
1.5.3	论文宣读	9
2	VAG 小组服务器	10
2.1	FTP 服务器	10
2.1.1	FTP 服务器主要目录	10
2.2	文件共享与打印服务器	11
2.2.1	文件共享服务器	11
2.2.2	打印服务器	12
2.3	SVN 服务器	12
2.3.1	SVN 服务器主要目录	13
2.3.2	SVN 服务器的使用指南	13
2.3.3	SVN 服务器的未来构建	14

2.4	论坛服务器	14
2.5	Wiki 服务器	15
2.6	VAG 主页	15
2.7	公共事务邮箱	15
3	VAG 小组人员信息	16
3.1	邮件列表	16
3.2	成员列表	16
4	共享资源	24
4.1	电子书籍	24
4.2	纸质书籍	25
5	日常事项与流程	27
5.1	毕业论文和答辩事项	27
5.2	离校流程	28
6	相关文献	29
6.1	相关文献介绍	29
6.1.1	主要文献介绍	29
6.1.2	文献查询方法	30
6.2	常用网址	32
7	书籍和常用学习资料介绍	33
7.1	C 语言学习	33
7.2	C++ 语言学习	33
7.3	图形学基础学习	34
7.4	OpenGL 学习	34
7.5	Direct3D 学习	34
7.6	CG 语言学习	35
7.7	GPU 编程学习	35

7.8	CUDA 编程学习	35
7.9	MFC 编程学习	35
7.10	QT 编程学习	35
7.11	编程杂书推荐	35
8	小丁讲编程	37
8.1	总论	37
8.2	Windows 下开发软件介绍	39
8.3	VS 插件介绍	39
8.4	Linux 下开发软件介绍	40
8.5	图形 API 介绍	41
8.6	常用库介绍(C/C++)	42
8.6.1	数学库	42
8.6.2	界面库	42
8.6.3	计算机视觉库	45
8.7	小叶来补充(C/C++)	45
8.7.1	可读性 V.S 效率	45
8.7.2	美好愿景:小组编码规范	45
8.7.3	继续介绍库	46
9	图形学需要的主要数学知识及资料	48
9.1	所需要的数学知识	48
9.2	搜集的资料说明	48
10	微分几何简介	50
10.1	曲线的曲率	50
10.2	曲面的曲率	52
10.2.1	Gauss 之前的工作	52
10.2.2	Gauss 的工作	53

11 科技论文写作杂谈	59
11.1 论文写作注意事项	59
11.2 导师在科技论文写作中的职责	63
11.3 关于选题和执行力	64
11.4 关于学术腐败和学术行为不端	66
11.5 How to have a paper get into SIGGRAPH?	68

1 VAG 小组日常制度

1.1 周报制度

周报是展示个人在过去一周的工作成果和心得,计划未来一周的工作进度的最佳平台。周报同时也是学生个人同导师之间进行工作和生活交流的一种有效方式。导师可以通过周报及时了解学生的工作进展。目前周报的提交方式推荐使用小组的 BBS 系统的 Weekly Report 板块。周报力求简明扼要,图文并茂。结合个人研究的实际情况,周报应当包括(但不限于)以下方面的内容:

- **课题相关论文阅读心得**。作为科研工作过程的重要一步,阅读与自身研究方向相关的论文有助于拓展我们自身的研究思路,发现当前该领域的主要热点问题和潜在研究课题,激发自己的研究灵感。阅读他人论文应关注论文所属的研究领域,论文所解决的研究问题,论文的主要创新点,论文提出的方法是否存在可改进的潜力以及能否直接或者间接被应用到自己的研究课题上来。
- **自身研究课题的进展**。在课题启动阶段,应简要介绍自身研究课题的背景和将要解决的问题和挑战,安排自己的总体研究进度,并列举若干具有代表性的相关前人工作。在课题不断深入的过程中,概述自己的研究思路和技术路线,通过比较实验结果,系统交互方式,计算速度(包括预计算速度和实时计算速度)以及内存占用等各个实验环节,说明自身研究方法的创新点和贡献。在课题收尾阶段,总结自身课题的研究成果,附上实验结果截图。如有原型软件,应当撰写简明软件使用指南。
- **学术活动**。学术界,学校,学院以及 CAD&CG 实验室会不定期地举行各类学术活动,包括各种学术会议,知名教授的报告演讲等。组员在参加这类之后,可结合本组的科研方向和情况,记录报告提供的相关资源,分享参加这类学术活动后的心得。

1.2 出勤制度

首先声明,我们 VAG 小组的出勤制度是宽松的,我们致力于建立自由、高效的学习工作环境。我们不提倡严格的出勤考勤制度,但是,我们要求 VAG 小组的所有成员尽可能的遵循如下一些出勤建议:

1. 我们建议早上 9:00 之前到达实验室,并确保工作时间内(周一至周五,9:00-17:00)的学习与研究效率。我们不提倡晚上离开实验室的时间晚于 22:00。除非工作需要,不提倡通宵。
2. 周一至周五时间内,如有特殊情况而未能出勤,必须确保小组内其他成员之一能够知道你的去向。
3. 周末属于自由支配时间,尽量保持良好的休息。积极参加小组组织的活动。如果工作需要,希望能调整好心态积极投入工作。
4. 特定情况下,请务必遵守导师所提供的出勤安排。

1.3 学术讨论制度

学术讨论主要形式为讨论班,即 VAG 小组的所有成员在一个固定的场所(一般为会议室)学习讨论相关领域的最新的学术文章,同时也讨论小组内成员的工作进展。讨论班是学生与导师进行交流的主要形式,也是小组内部进行学术交流的重要载体。讨论班是开放的,任何成员都可以发表成熟的或不成熟的想法。学术讨论制度规定所有成员必须准时参加讨论班,无故不得缺席。迟到或无故缺席都将被按照规定进行处罚。

讨论班按照参加人员的区别,主要分为大组讨论班和小组讨论班。其中大组讨论班要求 VAG 小组的所有组员都必须参加,有专人进行组织。小组讨论班主要按照课题方向进行分组,组员在短时间内相对固定。目前小组讨论班包括医学影像组、可视分析组和科学计算可视化组。

以下是一些大组讨论班的相关制度:

1. 讨论班每周举行一次,包括两个论文报告和一个工作报告;其中论文报告时间为 10~15 分钟,工作报告时间为 15 分钟;严格计时。
2. 其中前两位论文报告,必须选择当年的 **VisWeek** 最新论文;请务必提前两周准备,自己精读、理解论文;报告时间为 10~15 分钟,讨论时间为 5 分钟。
3. 其中工作报告内容为报告人在本阶段时间内的的工作进展,时间为 15 分钟。
4. 博士生应该严格要求自己,认真准备论文报告,听报告时积极发言,形成良好的学术氛围。
5. 大组讨论班将安排一个可选听的环节(非特殊情况,博士生必须参加并发言),时间在 30 分钟以内,针对热点研究、科普、自我修炼等方面话题研讨。每次由一个二年级以上博士生主持,提前 2 周确定讨论话题并通告所有讨论班参加人员。请所有参与人员思考、自行阅读材料。主持人在开始后用两到三分钟报告一个汇总观点的 **Slides**,然后开展讨论。
6. **VAG** 小组所有成员必须准时参加讨论班,因故缺席需向导师请假。
7. 讨论班负责人在每次讨论班结束之前,通知下次讨论班的举行地点、报告人员。讨论班结束之后,报告人员需及时将报告内容相关的文件上传至 **FTP** 服务器相关目录,讨论班负责人需要及时进行审核。
8. 报告人员需提前 3~4 天将报告论文或项目的题目及报告内容概要通知讨论班负责人,并将报告论文上传至 **FTP** 服务器相关目录;讨论班负责人通过电子邮件告知讨论班所有参加人员下次讨论班的举行地点、报告人员及讨论班内容。
9. 报告人员需提前 10 分钟准备好报告材料,并完成设备调试。
10. 讨论班以半年为一个时间周期,按照学校春、夏、秋、冬学期进行相应调整。讨论班负责人按半年为周期,提前安排每次讨论班的报告人员。
11. 由专人负责记录讨论班内容(包括可选听环节),并在整理后通过 **wiki** 和邮件分享于所有可视分析组成员。

☞ 目前大组讨论班举行时间为每周五晚上 6:30 至 8:30,由夏菁负责组织。大组讨论班由严炳辉负责记录,各小组讨论班内容记录人员分别为:医学影像组由徐星负责记录,可视分析组由夏菁负责记录,科学计算可视化组由李昕负责记录。

☞ 目前选听环节主持人按照以下顺序轮换:
陈海东,彭帝超,陈伟锋,夏菁,丁治宇,刘真,李昕。

1.4 经费报销制度

1.4.1 经费来源

VAG 组的经费来源为各个纵向项目,即来自于全国纳税人,课题结题时将由国家专业人员审计,因此必须严格按照预定经费比例使用。组员应理解财务制度,优化使用经费,实现合理、合法、高效的发票来源要尽量多元化。

表 1.1 VAG 组的经费来源主要包括(目前没有横向项目):

	资助方	验收难易程度	易用程度
973 项目	科技部	非常严格	容易
863 项目	科技部	非常严格	不容易
国家自然科学基金	基金委	不严格	非常容易
浙江省自然基金	省基金	不严格	非常容易
浙江省公益项目	科技厅	严格	容易
重点实验室运行费	教育部	国库支付,即日审计	容易
横向项目	企业	不严格	非常容易

注:为防止贪污、节约经费,国家级、省级部分项目已经或将采用国库支付的方式,即经费不再拨往承担单位,而是银行,经费使用采用即报销即审计的方法。经费到期未用完即收回。

固定资产购买、入库流程:

购买硬件 → 获得发票 → 到实验室管理员(李伟青老师)处办理固定资产入库手续 → 在固定资产托管人(陈为)的浙大设备管理信息系统帐户中,登记并认领硬件,须指明硬件所在地 → 报销后,获得以出售单位为题头的支票,再由出售单位自行取现。

注:部分经费要求硬件购买是采用招标方式进行。

国际交流费用:

研究生或老师必须单独填写浙江大学出国申请表(在浙江大学外事处网站下载),经过实验室、学院、外事处审批后,还需要到科研院查询资助项目是否有足够的

表 1.2 纵向经费的可使用类别和比例包括(横向项目经费比例非常灵活):

类别	973	自然	863	实验室	发票明目	备注
人工	8%	15%	<25%	15%	研究生培养费	只用于参与课堂研究的人员
设备	15%	15%	<20%	0%		指用于科研的 单件 价格 800 元以上的硬件设备,必须做成固定资产并在网上入库
材料	15%	15%	<20%	15%	电脑配件、办公用品、电脑耗材、墨盒、光盘、复印纸	所有属于办公用品、电脑配件的物品,都可以归于材料
差旅	15%	15%	<20%	15%	火车票、汽车票、出租车、飞机票	用于科研目的如会议、研讨会、调研
会议	5%	0	5%	5%	专家验收与研讨	项目的专家参与项目研讨
国际	10%	15%	15%	0	出国会议、国外来访	需要事先申请出国审批备案、科研院经费查询
传播	10%	10%	10%	10%	论文出版、会议注册、通信、邮寄、书本、冲印、网络	预付费发票不能报销,如移动充值的预付费
测试	10%	10%	10%	10%	实验试剂、加工	需要展现测试加工的需求
餐饮	0	0	0	0		只能在横向课题中报销

国际交流费用。973、863 等项目在申请时需要注明将出访的国家和会议,如果不符合,可能被科研院拒绝。另外,如果老师不属于 973、863 项目申请团队,也不能被资助 ---研究生不属此列。

1.4.2 报销制度

部分组员的劳务费通过报销的形式发放。通过报销金额于每月 9 日以后的周五下午打入浙江大学分配的个人银行卡帐户。通常情况下,硕士生和博士生打入银行卡的金额不超过 800 元,超出部分以现金发放。本科生、在职博后等没有浙江大学银行卡的人员则直接通过现金形式发放。

每学年,有专门人员负责报销制度,负责收集发票、维护报销名单、发票整理、填写报销单(分为差旅单、日常用品单、国际费用单)、向导师汇报等。

VAG 组的报销制度相当弹性,发放金额与上交发票金额没有绝对联系。每个月月初收集发票一次,统计整理后交由陈老师签字,交给金叶英去实施报销。根据发放的上限和个人在当月的表现发放劳务费。超过实际发放的那部分发票会在发票池内累积,以平衡今后可能出现的发票不足。

我们鼓励贡献发票,大家贡献的发票报销来的金额将累积入小组的公共经费。

1.4.3 公共经费

小组公共经费是由陈老师率先拿出 10000 元启动资金所建立的资金池。小组公共经费是组内同学各项福利的来源,如食品的供应,购物卡的发放,聚餐经费,小组公共硬件、图书、运动器材的采购,会议差旅费用等大额资金的周转等。

目前小组经费的增值主要还是靠平时采购时多开的发票。例如缘网采购时,可以多付税点,尽量多开一些发票。也抓住一切其他机会,如某同学要买手机等大件,一般淘宝上的商家在付税点的情况下可以开发票,那就要抓住机会,付税点,弄发票。

有一个专门存放小组经费的中国银行账户和银行卡,为陈老师提供。这张经费卡由管理的同学掌管。在拿到报销的小组经费后,要及时将大额现金存入经费卡,少量现金可随身携带,以备小组使用。

在 VAG 论坛里专门有一个管理板块,其中一个帖子是专门用于记录小组经费的收支的。负责财务的同学每次接任后可开帖记账:

<http://zjuvag.3322.org/bbs/viewthread.php?tid=235&extra=page%3D1>

小组的经费需专款专用,用于采购集体所需的物品。其他例如个人更新非必需的硬件都不得动用小组经费,除非经陈老师同意使用小组经费。

在每年极端气候情况下,如暴风雨、大雪、酷热,可以用公共经费为大家订购盒饭。另外,每年举办活动也从小组公共经费开支。

1.4.4 发票报销

发票可以在平时就开始收集,大家一有发票就交给负责同学,负责同学在月末时做统计,参看以往的电子表格。在发票后面签字后,一起交给陈老师。除出租发票和车票之外的其他发票,背面都需两个人的签名,即:出租车发票和车票一个签名即可。

切记所有单张发票的金额不能超过 800, 包括 800 也不行。超过的只能算作固定资产, 操作非常繁琐。

在报销过程中差旅发票和其他发票是分开两张表格的, 所以分类时要将这两类分开。

报销的一些要点:

- 所有的发票抬头必须是浙江大学
- 市内交通包括市内的公交车发票、出租车票、由市民卡公司发售的公交卡报销凭证(小白纸)
- 每月从移动公司打印的手机话费单
- 住宿发票必须配合入住当天的车票或机票, 且火车、汽车、飞机必须有返程发票。
- 通常情况下, 杭州市本地的住宿票很难报销。
- 每年第一季度可以报销前一年的发票, 之后只能报销当年的发票。发票尽量不要囤积。
- 浙江大学网络 VPN 费用可以开具红色的校内收据, 可以作为通信费用报销。但是食堂开具的红色收据只能在横向发票中报销。
- 不能报销的发票类型: 餐饮、汽油、停车、旅游、儿童车票、假发票等。

小组的采购尽量去能开我们能报的发票的地方, 例如京东商城、亚马逊、易迅, 缘网商城, 部分淘宝店也可以开具发票, 需要询问。发票抬头为浙江大学, 内容为办公用品或者电脑配件。发票金额不超过 800, 当需要采购 800 以上的物品时, 发票要分开开。发票不能连号, 跟商家说, 尽量不要连号的。缘网是校内做类似的生意较多, 懂得这方面的规矩, 消费类的券和超过 800 的物品采购首选缘网。

京东等网上商城购物保修一般需要发票复印件, 因此在收到物品后, 要先用相机拍摄发票留底, 再进行报销。

1.5 科研工作展示

作为研究生(博士和硕士),我们在实验室的主要工作是项目和研究,一般而言研究也是为项目服务的。上述各种制度是我们 VAG 小组作为一个整体而需要遵守的一个规范,制定这些规范的目的就是促使我们能够按时保质量地完成项目并发表成果。因此本章节中,我们将讨论下关于科研工作展示的相关内容。

我们知道,与诸位关系最紧密的,特别也是诸位新成员所关心的,应该就是写论文与发论文了。一般来说,论文主要分为两个阶段,即论文准备阶段与发表阶段。如果你投的是会议论文并且被会议接收,则还有一个论文的宣读阶段。在每一个阶段,我们都必须全力以赴,做好一切可能为论文添砖加瓦和润色的工作(可能是非常枯燥的)。

1.5.1 论文准备

论文从一个 idea 开始,逐步经过优化、改进,最终得到一个可以发表的结果后,将相应过程整理成文并对文字进行修饰润色;对于一些需要表达交互效果的工作,最好还需要制作一个视频或发布一个应用程序。这就是一个论文的产生过程。

通常,论文的写作与论文所述工作的实现是并行进行的。在工作进行过程中,每一个记录(周报、日记等)都可能会成为论文的一个章节,因此建议大家一定要重视记录实现中的每一个细节。

1.5.2 论文发表

论文投稿以后,需要经过一个漫长的审稿时期。对于会议论文,从论文接收到会议举行会有一个明确的时间表。而对于期刊论文,审稿时期长短差异很大,需要你经常关注投稿系统中的状态变化。

论文被通知接收后,一般都会有多个审稿人的意见,即我们通常所说的 minor/major revision。后者说明论文的想法被认可但是表达和实现结果没有达到可以被发表的要求,因而需要被大改。因此,我们需要根据这些审稿意见对论文进行修改,有时甚至需要做更多的结果。对于审稿人提出的每一个问题,我们都必须一一作答并写入到修改后的论文中。如果问题无法回答或无法被解决,也一定要和审稿人进行

沟通交流(一般投稿系统中都有电子邮件交流的方式)。**这一阶段的工作是一篇文章发表过程中最重要、也是最关键的**,稍有不慎,可能会导致前功尽弃,因此必须认真对待。一般而言,只要认真回答了审稿意见中的每一个问题,论文被发表就基本十拿九稳了。

修改后论文提交后,不久应该会有第二次审稿结果。此时可能仍然需要进行大量的修改,但绝大多数是限于文字和格式层面的,因此工作会相对轻松些,但可能有点枯燥。另外这时还需要填写一些版权转移申明等文件,并完成版面费(若需要)的转账等事情。

1.5.3 论文宣读

对于被录用的会议论文,通常被要求在大会上进行宣传。宣读论文的准备工作的主要是 PPT 的制作与准备、海报制作(具体根据大会的要求而定)。一般论文的宣读时间在 15-20 分钟,参会的报告者必须提前做好各种准备。宣读后一般会留一定时间给其他参会者提问。论文展示或宣读后,同行研究者如果对你的研究感兴趣,通常会在茶歇、就餐等时间同你进行交流。

对于被录用的会议论文,通常大会委员会要求至少有一名作者在大会上宣读该论文,因此也需要至少一名作者参加大会。开会进行交流的重要手段,因此只要经费允许,导师是绝对支持的。开会手续主要包括会议注册,注册费支付,宾馆交通安排等事项。国际会议还包括护照签证、机票预订等手续。学校研究生院对每一位研究生在学期间提供一次赴海外参加国际学术会议资助,具体参考《浙江大学研究生赴海外参加国际学术会议资助条例》。

这里要稍微提醒下,不是所有期刊或会议都能得到导师的支持,因此在投稿前需要了解目标期刊和会议的级别,影响因子等情况,并经过导师同意后方可投稿。

必须注意:每篇论文的投稿和发表都必须经过老师同意,并且在发表时在填写基金资助方面的内容时征询老师意见。

2 VAG 小组服务器

VAG 小组拥有一台专用台式机作为服务器,在这之上架设了 FTP 服务器、文件共享与打印服务器、SVN 服务器、论坛服务器以及 Wiki 服务器(建设中)等。其中文件共享服务器被设置为只读的,与 FTP 服务器使用相同的目录,方便用户直接查看文件内容。服务器的 ip 已绑定了动态域名:zjuvag.3322.org。若今后 ip 地址有变动请邮件联系 vag.zjucadcg@gmail.com,将有专门负责人进行修改。

2.1 FTP 服务器

FTP 服务器主要用途为小组公共数据的存档以及个人数据和文档的交换中心。目前地址为 <ftp://vag:vag@zjuvag.3322.org:21>。需要注意的是,帐号 vag 是全权限帐号,因此请**谨慎对待删除、覆盖等操作**,也尽量不要让非小组成员知道此 FTP 的存在。

2.1.1 FTP 服务器主要目录

FTP 主要目录结构安排如下:

1. FTP2/Data 目录

Data 目录用于存放各类实验原始数据,包括 CT Data、DTI Data、Time-Varying Volume Data 以及一年发布一次的 IEEE Visualization/InfoVis/VAST Contest Data。该目录由专人负责管理,其他人请不要上传数据到此目录。

2. Incoming/E-Book 目录

E-Book 目录用于存放和分享各类与小组研究方向相关的电子书籍,鼓励大家将自己收集到的优秀电子书籍上传到该目录,使小组中有阅读需要的同学方便快速地定位此类资源。

3. Incoming/Seminar 目录

Seminar 目录用于归档每周小组讨论班上的报告,演讲资料。每一周,请有报告

任务的组员以日期为子目录名,提前将报告资料(包括 PPT、PDF 等)上传至该目录下。

4. Incoming/Software 目录

Software 目录主要用于存放各类常用软件的安装文件,特别是文件尺寸较大的大型软件。对这类安装文件进行存放,使组员能够利用实验室局域网高带宽的条件快速地下载此类资源到本地机器,提升效率。鼓励大家将最新最全的软件资源上传到该目录,并及时删除老版本的重复资源。

5. Incoming/Users 目录

每个 VAG 组员都拥有权限 Users 目录下新建一个以自己中文全拼为名字的子目录,用于存放个人的数据和文档。注意,请尽量不要上传与学习和研究无关的数据,例如电影,音乐和游戏,并定时整理自己的子目录,删除过期的资源。

6. Incoming/Temp 目录

Temp 目录用于临时的数据和文档存放。该目录下的资源由专人定期进行清理,清理之前不进行通知,希望每个人都在本地机器上做好数据的备份。

2.2 文件共享与打印服务器

2.2.1 文件共享服务器

FTP 服务器不支持文件的在线浏览与播放,如果需要查看文件内容,我们必须把文件下载到本地。为了更加方便的查看文件内容,我们在局域网内部开放了一个只读的文件共享服务。我们推荐将远程目录映射为一个网络驱动器使用文件共享服务。具体操作如下:

1. 打开“我的电脑”,选择“工具”->“映射网络驱动器”;
2. 将远程目录“\\VAG-SERVER\ftp”填入后完成。

使用文件共享服务可以方便的查看远程目录上的文件。例如,我们 FTP 服务器中包含了往年 VisWeek 的光盘镜像,它们通常比较大,使用文件共享服务后我们可以

直接通过 **Daemon Tool** 等软件挂载并使用这些光盘镜像而不用将它们拷贝到本地磁盘上。文件共享服务主要是方便我们对 **FTP** 服务器上的文件内容和目录的查看,如果你将频繁地使用某些文件,我们仍然会建议你直接将这些文件拷贝到本地磁盘上。

2.2.2 打印服务器

VAG 小组拥有自己的打印机,打印机安装在服务器上。要使用打印机,你首先需要在本地机器上安装打印机驱动,具体操作如下:

运行“Win+R”,输入“\\VAG-SERVER”,双击“打印机”要求安装驱动,安装完驱动后会在本地的打印机列表中看到一台远程打印机。

由于打印机属于小组资源,因此请尽量节约用纸用墨。使用打印机的几点建议:

1. 一般阅读的 **paper** 尽量使用双面打印;
2. 尽量避免打印书籍;
3. 尽量避免打印深色背景的页面,如 **PPT** 等文件,可以修改背景后再打印;
4. 打印机没墨后,请主动将纸盒取出,并留言说明和联系打印机管理员。

2.3 SVN 服务器

SVN 服务器主要面向软件开发和论文撰写管理提供版本控制和同步的服务。目前地址为 <svn://zjuvag.3322.org>,每个组员的姓名全拼作为自己的用户名和密码,小组新人请通知组内管理员师兄添加账户。在使用 **SVN** 服务前,请先安装 **SVN** 客户端 **TortoiseSVN**。

需要注意的是,尽量避免临时文件、中间文件等的提交,这些没有备份意义的文件只会起到浪费服务器空间的作用。(比如程序工程中的 **Debug**、**Release** 目录里所有的文件,再比如 **Latex** 编译产生的临时文件 ***.aux**、***.log** 等这些文件都不需要提交到 **SVN**)

另外,在多人协同合作时,单一时段内数据文件应只由一个人更新,等到此人 **Commit** 以后,其他人首先进行 **Update**,然后进行再进行编辑更新,这样可防止协作过程中数据版本的冲突。

2.3.1 SVN 服务器主要目录

SVN 主要包含四个大目录：

1. Papers 目录

Papers 目录主要存放本组多人协同合作的科研投稿论文相关的所有数据。子目录均以课题名命名, 包含投稿论文的 **Latex** 源码, 所有插图, 视频 **Demo** 等数据。

2. Projects 目录

Projects 目录用于存放多人协作开发的原型软件系统的开发源码和进行版本控制。建议不要将大量原型软件系统的原始数据文件和产生的中间数据文件在此处进行存档。原始数据文件和中间数据文件可在 **FTP** 服务器上进行存档。

3. Personal 目录

Personal 目录用于个人开发的原型软件和科研论文的源码和相关数据的存档和版本控制。每个组员拥有权限在该目录下创建以自己名字全拼命名的子目录。在存档时, 注意合理利用服务器空间, 定期删除不必要的数据和文件。

4. Homework 目录

Homework 目录用于存档各门研究生专业课程的最终大程, 是低年级的师弟师妹了解各个大程具体要求, 快速进行大程开发的有效渠道。建议各位组员为自己的大程撰写简要的用户指南, 包括界面说明, 关键数据结构和算法流程。

2.3.2 SVN 服务器的使用指南

由于目前我们的 SVN 服务器只使用了一个 **repository** 进行版本控制, 因此某一个错误的操作可能会影响到整个 **repository** 的安全。因此有必要在本指南中简单介绍 SVN 服务器的基础使用。

1. 安装 SVN 客户端 TortoiseSVN, 下载地址为<http://tortoisesvn.net/downloads.html>。

2. 在桌面或文件夹的任意位置点击右键,选择“TortoiseSVN->Repo-browser”打开一个 Repository browser 窗口。
3. 从 Repository browser 窗口左侧定位到2.3.1章节提到的主要文件夹结构的相应位置,在右侧窗口空白处点击右键,选择“Create folder...”新建一个文件夹。
4. 选择刚才新建的文件夹,右键选择“Checkout...”,然后设定本地计算机上导出的目标文件夹即可。此时本地计算机上的文件夹已经受到 SVN 服务的版本控制。

以上仅仅是使用我们 VAG 小组 SVN 服务器的一般步骤,更多的功能及操作请参考 TortoiseSVN 帮助文档。需要注意的是,你所得到的并使用的 SVN 账号(姓名全拼)是全权限的,因此请确保你的所有操作尽量限制在 SVN repository 的个人文件夹内部或者你所参与的项目文件夹内部。除非你是管理员,否则请不要对其他文件夹进行修改、移动或删除等操作。

2.3.3 SVN 服务器的未来构建

在使用 SVN 服务的过程中,你可能会发现你正在进行的项目的版本号控制不一定是连续的,这主要是因为目前我们的 SVN 服务器仅使用了一个 repository(SVN 服务的存储仓库)。由于我们实际使用的是这个 repository 中的一个文件夹目录,因此不连续版本号不会影响到我们对正在进行项目的版本控制。为了更好更安全地使用 SVN 服务器资源,我们期待建立一个完善的 SVN 服务系统。

2.4 论坛服务器

本组拥有一个内部的专用论坛,目前地址为<http://zjuvag.3322.org/bbs/>。论坛用途涵盖:本组各类管理信息的记录和发布,组员周报的发布,各个科研课题的过程记录和管理,各类计算机资源和技术新闻的通知,本组课余生活和组员风采的真实记录等。

论坛超级管理员为陈为老师。日常论坛管理工作由专人负责,应做到每周备份服务器数据和用户信息一次。鼓励每位组员经常登陆论坛查看最新小组动态,提倡申请成为论坛各版面的版主,积极发帖,热烈讨论,营造小组独有的内部文化氛围。

2.5 Wiki 服务器

在部署本指南时,Wiki 服务器还处于婴儿时期,希望各位组员积极为该 Wiki 输送养分,将该 Wiki 打造为造福 VAG 的在线百科。目前地址为<http://zjuvag.3322.org/wiki/>。Wiki 的内容包括但不限于以下内容:

- 可视化各领域经典概念解释;
- VAG 组内文化宣传;
- 有参考意义且适于总结成专题的编码(C/C++/Java/Latex/...)技巧或提示。

2.6 VAG 主页

不知大家注意了没,论坛和 Wiki 使用的是“bbs”和“wiki”目录,那么<http://zjuvag.3322.org/>上面有什么? 嗯,应该是 VAG 自己的主页,不过撰写本文时此主页还没出生,希望有热情的组员勇敢的承担起“主页爹妈”的岗位。有了主页,VAG 在 zju 校内的宣传又多出了一个途径。

2.7 公共事务邮箱

我们小组维护一个用于处理公共事务的邮箱,地址为vag.zjucadcg@gmail.com。当服务器未能正常工作、个人信息需要更新、需要添置硬件软件等,或您认为本指南有需要修订时,都可以直接联系此电子邮件,将由专门负责人负责处理这些事务。另外,本邮箱的另一重要使命是负责我们 VAG 小组的对外交流,包括外校学生联系等。

3 VAG 小组人员信息

3.1 邮件列表

为便于管理及方便大家联系,我们有如下一些邮件列表。当需要进行讨论或通知的时候,可以选择相应的邮件列表进行发送。如果有成员需要加入或者退出,请联系各组的管理人员。([各组目前的成员列表将在近期更新](#))

组	邮件列表	负责人
大组	zjuvag@googlegroups.com	夏菁
气象组	climate-vag@googlegroups.com	严炳辉
三维可视化组	scivis-vag@googlegroups.com	陈伟锋
集群可视分析	ali-vag@googlegroups.com	夏菁
信息可视化	infovis-vag@googlegroups.com	彭帝超
医学	medvis-vag@googlegroups.com	徐星

3.2 成员列表

本节将维护一个 VAG 小组所有成员的列表,包含所有成员的联系方式。已毕业成员信息正在收集中。为确保时效性,本章将每个季度更新一次。

导师



姓名: 陈 为
 手机: 13588477530
 主页: www.cad.zju.edu.cn/home/chenwei
 Email: chenwei@cad.zju.edu.cn
 工作单位: 浙江大学 CAD&CG 国家重点实验室
 MSN: shearwarp@hotmail.com QQ: 1317265502

博士后

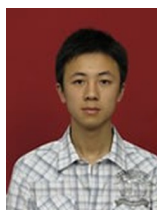


姓名: 刘真
手机: 13735514271
主页:
Email: liuzhen@zjucadcg.cn
工作单位: 浙江大学 CAD&CG 国家重点实验室
MSN: mimulz@hotmail.com QQ: 8495693



姓名: 彭帝超
手机: 15157180966
主页:
Email: pengdichao@cad.zju.edu.cn
工作单位: 浙江大学 CAD&CG 国家重点实验室
MSN: dichao@gmail.com QQ: 13871255

博士研究生



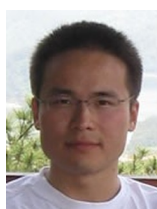
姓名: 陈海东 (2011-)
手机:
主页: <http://www.cad.zju.edu.cn/home/chenwei/vag/chd/index.htm>
Email: chenhd925@gmail.com
工作单位: 浙江大学 CAD&CG 国家重点实验室
美国 Bosch 研究院北美中心联合研究计划 (2010.8-2011.8)
MSN: chenhd925@hotmail.com QQ: 125681394



姓名: 丁治宇 (2011-)
手机: 13567135550
主页:
Email: dingzhiyu@zjucadcg.cn
工作单位: 浙江大学 CAD&CG 国家重点实验室
MSN: QQ: 395491125



姓名: 夏菁 (2011-)
手机: 13732223076
主页:
Email: summer@zju.edu.cn 或 jjane.summer@gmail.com
工作单位: 浙江大学 CAD&CG 国家重点实验室
MSN: summer179279@hotmail.com QQ: 178279172



姓名: 陈伟锋 (2009-)
手机: 13067737953
主页:
Email: chenweifeng@cad.zju.edu.cn
工作单位: 浙江大学 CAD&CG 国家重点实验室
MSN: tonycharce@hotmail.com QQ: 104810892



姓名: 李昕 (2009-)
手机: 13805784243
主页:
Email: lixin@zjucadcg.cn
工作单位: 浙江大学 CAD&CG 国家重点实验室
MSN: isabelincoln@gmail.com QQ: 6253915

硕士研究生



姓名: 解 聪 (2011-)
手机: 13989823428
主页:
Email: xiecng@gmail.com
工作单位: 浙江大学 CAD&CG 国家重点实验室
MSN: juliux@live.cn QQ: 840164011



姓名: 徐 星 (2010-)
手机: 13968013088
主页:
Email: xuxing211@gmail.com
工作单位: 浙江大学 CAD&CG 国家重点实验室
MSN: xx_starry@hotmail.com QQ: 63292639



姓名: 严丙辉 (2010-)
手机: 15868192431
主页:
Email: ybh.zju@gmail.com
工作单位: 浙江大学 CAD&CG 国家重点实验室
MSN: ybh_cg@hotmail.com QQ: 623854078



姓名: 朱斯衍 (2010-)
手机: 1(650)5615013
主页: <http://www.cad.zju.edu.cn/home/chenwei/vag/zsk/index.html>
Email: eziasolsky@hotmail.com
工作单位: 浙江大学 CAD&CG 国家重点实验室
美国 Bosch 研究院北美中心联合研究计划 (2011.5-2012.5)
MSN: eziasolsky@hotmail.com QQ: 752032750



姓名: 陈广宇 (2009-)
手机: 18668000116
主页:
Email: mail2cgy@gmail.com
工作单位: 浙江大学 CAD&CG 国家重点实验室
MSN: chenguangyu@zju.edu.cn QQ: 522121147



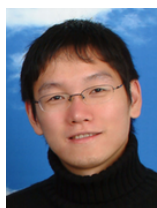
姓名: 王桂珍 (2009-)
手机: 18801132648
主页:
Email: wangguizhen@zjucadcg.cn
工作单位: 浙江大学 CAD&CG 国家重点实验室
微软亚洲研究院访问学生 (2011.6-2012.3)
MSN: randyef@163.com QQ: 492741737

本科生



姓名: 吕 能(2008-)
手机: 15088715799
主页:
Email: freeneng@zju.edu.cn
工作单位: 浙江大学 CAD&CG 国家重点实验室
MSN: freeneng@hotmail.com QQ: 392402510

已出国、毕业或工作



姓名: 邹林灿 (硕士)
导师: 彭群生、陈 为
主页:
工作单位: 美国 Bosch 研究院北美中心 (2010-至今)
美国 Bosch 研究院北美中心联合研究计划 (2007.6-2008.6)



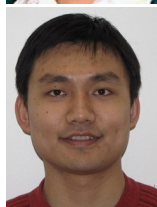
姓名: 梁 潇 (硕士)
导师: 彭群生、陈 为
主页:
工作单位: 美国微软亚洲研究院 (2007-至今)



姓名: 谈奇峰 (硕士)
导师: 彭群生、陈 为
主页:
工作单位: 美国谷歌公司中国分部 (2007-至今)



姓名: 龚 恻 (硕士)
导师: 彭群生、陈 为
主页:
工作单位: 美国加州大学圣巴巴拉分校博士研究生 (2006-2012)



姓名: 董 朝 (硕士)
导师: 彭群生、陈 为
主页:
工作单位: 美国康乃尔大学 (德国马普所博士毕业) 博士后 (2005-2012)



姓名: 夏佳志 (硕士)
导师: 陈 为
主页:
工作单位: 新加坡南洋理工大学博士毕业 (2007-2011)



姓名: 严志程(硕士)
导师: 陈 为
主页:
工作单位: 美国伊利诺伊香槟分校博士研究生 (2010-至今)
美国 Bosch 研究院北美中心联合研究计划 (2009.7-2010.7)



姓名: 丁子昂(硕士)
导师: 彭群生、陈 为
主页:
工作单位: 美国普度大学博士研究生 (2010-至今)



姓名: 于 洋(硕士)
导师: 彭群生
主页:
工作单位: 美国罗特格斯大学博士研究生 (2009-至今)



姓名: 曾 运(硕士)
导师: 彭群生
主页:
工作单位: 美国纽约州立大学石溪分校博士研究生 (2006-至今)



姓名: 范亦楠(硕士)
导师: 彭群生
主页:
工作单位: 美国德州农机大学博士研究生 (2006-至今)



姓名: 张钰勃(本科)
导师: 陈 为
主页:
工作单位: 美国加州大学戴维斯分校博士研究生 (2010-至今)



姓名: 骆鹏程(本科)
 导师: 陈 为
 主页:
 工作单位: 美国加州大学戴维斯分校博士研究生 (2007-至今)



姓名: 傅靖华(本科)
 导师: 陈 为
 主页:
 工作单位: 美国北卡教堂山分校博士研究生 (2010-至今)



姓名: 吴 岚(本科)
 导师: 陈 为
 主页:
 工作单位: 加拿大大不列颠哥伦比亚大学博士毕业 (2005-2008)



姓名: 陈 曦(本科)
 导师: 陈 为
 主页:
 工作单位: 加拿大大不列颠哥伦比亚大学博士研究生 (2007-至今)



姓名: 方 菁(本科)
 导师: 陈 为
 主页:
 工作单位: 加拿大大不列颠哥伦比亚大学博士研究生 (2010-至今)



姓名: 胡越琦(本科)
 导师: 陈 为
 主页:
 工作单位: 美国 Bosch 研究院北美中心联合研究计划 (2010.5-2011.5)



姓名: 温朝侃(硕士)
 导师: 陈 为
 主页:
 工作单位: 创业



姓名: 叶 榭(硕士)
 导师: 陈 为
 主页:
 工作单位: 北京www.sogou.com

4 共享资源

为更好的共享 VAG 小组内部的一些资源,本手册将一些重要的如各类电子和纸质书籍、相关软件、重要报告等可视化领域的一些重要资源在此略作不完备的汇总,供各位参考。

4.1 电子书籍

☞ 以下电子书籍主要存放于 FTP 服务器 (2.1.1) 的相关目录。考虑到版权关系,请各位组员勿将这些电子书籍在小组外部或网络上进行传播!!

1. Computer Vision: Algorithms and Applications
2. Data Mining Concepts and Techniques 2nd Edition
3. The Visualization Handbook
(此乃可视化经典入门教材,总体而言语言是通俗易懂的)
4. Visual Thinking for Design
5. Topological Modeling for Visualization
6. Efficient Visualization of Tensor Fields with Application to Magnetic Resonance Data
7. Beautiful Visualization Looking at Data through the Eyes of Experts
8. Information Visualization Perception for Design
9. Visual Displays of Information A Conceptual Taxonomy
10. Beautiful Data visualization in medicine
(医学小组成员入门教程)

以下是大神 Edward Tufte 之作:

1. Data Analysis for Politics & Policy
2. Envisioning Information
3. Tufte Beautiful Evidence lo-res
4. Visual and Statistical Thinking
5. Tufte Visual Display of Quantitative Information low-res

4.2 纸质书籍

📖 以下纸质书籍目前主要存放于服务器所在的物理位置。作为公共书籍,请注意合理使用,切勿在纸张上做过多的记录,并在阅读完毕后放回原处。

1. Visualization Toolkit
(vtk 的英文教材)
2. Information Visualization --- Perception for Design 2nd Edition
(Colin Ware)
3. Information Visualization --- Beyond the Horizon 2nd Edition
(Chaomei Chen)
4. Visual Thinking of Design
(Colin Ware)
5. Interactive Data Visualization --- 1st
(Matthew Ward, Georges Grinstein, Daniel Keim, A K Peters)
6. GPU Gems
(Randima Fernando, Addison Wesley)
7. 超网络理论及其应用
(王志平, 王众托)

8. 超图的理论基础
(王建方)
9. 超图 -有限集的组合学
10. 可视化数据
(Ben Fry 著, 张羽译)

5 日常事项与流程

5.1 毕业论文和答辩事项

学院对于硕士生毕业没有强制的论文要求,若已发表 EI 检索以上学术论文,则可以不经过程预审直接在论文提交截止日期前提交论文到研究生管理网。没有发表 EI 检索以上学术论文的同学则需按照学院要求提早提交毕业论文到各自的实验室、研究所进行预审。预审通过后可以进入答辩申请流程。

预审通过者需带预审表和打印的毕业论文,未经预审者需带发表论文的相关材料和打印的毕业论文,到学院研究生科段建平老师处审核。通过后,研究生管理网上即可以提交答辩申请。具体流程可参照研究生网上的办理流程。

毕业论文预审、答辩阶段的酬金可以报销:

1. 需下载并打印酬金表,请收取酬金的专家签字
2. 答辩完成后导师签字,带答辩记录表、表决票等材料去学院盖章
3. 去实验室小金老师处领取经费折子去学校财务处报销

本组秉持 VAG 出品,必属精品的原则,每年出产的硕士生数量都很少,保持每年 1 到 2 名的速度。因此,硕士生答辩需要与其他老师的学生组团,4 人以上较为合适。

答辩准备:

1. 酬金:答辩时将酬金装入信封,随论文打印稿一起交于答辩专家。
2. 答辩意见:需提前准备好答辩意见,经导师修改后打印好,答辩通过时需要宣读。
3. 水果:答辩前购买一些水果干粮供答辩专家食用。

答辩后工作:

1. 以管理员身份登录研究生管理网,录入答辩评语。
2. 将答辩记录表、表决票等材料提交研究生科审核。
3. 审核通过之后,即可开始离校流程。

(感谢[陈广宇](#)同学的分享)

5.2 离校流程

按照电子离校单系统<http://appl.sdc.zju.edu.cn/lxdxt/>里的指南, 通过所有审核后即可离校。

离校前工作:

1. 明确离校日期
2. 工作整理: 结束手头工作, 撰写文档, 录制 **Demo**, 将所有相关材料集中归档, 上传至 **ftp** 或者 **svn**。
3. 工作转交: 离校前将未完的工作交由师弟师妹, 确保接手者完全掌握包括操作、代码、原理等方面。将未完工作写成文档, 转交接手者。
4. 论文上传: 将毕业论文、答辩 **ppt**、相关代码文档, 上传至 **ftp** 的 **thesis** 目录下。
5. 个人物品清理: 离校前将实验室设备材料归还, 计算机设备个人资料清空, 用户名密码等转告接手者。柜子里的个人物品需带走。
6. 最后, 别忘了穿上学位服, 来实验室跟老师同学合个影, 留下记忆。也别忘了以后多多回来实验室交流访问。

(感谢[陈广宇](#)同学的分享)

6 相关文献

6.1 相关文献介绍

6.1.1 主要文献介绍

可视化与图形学所需要的参考文献很多,且不光局限于图形学本身。这主要表现在,不论可视化还是图形学,都是为了以可视的方式展现各领域的内容,因此,往往以其他领域的知识为基础,并把这些知识应用到我们的工作当中。这其中包括心理学,力学,以及生物,医学等方面的文献。在本节中,我们主要介绍一些属于本领域的杂志及会议。

首先,介绍可视化方面的会议。这方面有很多会议,最主要的包括 IEEE Visualization (Vis), Eurographics/IEEE Symposium on Visualization (EuroVis), IEEE Pacific Visualization Symposium(Pacific Vis)。其中,Vis 是可视化方面最主要的会议,在美国举办,开始时只是一个单独的会议,之后又逐渐增加了与之同时举行的 IEEE Information Visualization(InfoVis),IEEE Symposium on Visual Analytics Science and Technology (VAST) 两个会议,分别在可视化的基础上,更偏向于信息可视化和可视分析。整个会议被扩充为一周,被称为 VisWeek。之后,EuroVis 也是可视化的著名会议,每年在欧洲举行。Pacific Vis 是近年来开始举办的可视化会议,在亚洲举办。

在图形学方面,最主要的会议是 SIGGRAPH,每年夏天在美国举行,集中了每年图形学最高水平的文章,同时,近年来也发展成为工业界和学术界交流的平台,大部分与图形相关的企业都会来参加 SIGGRAPH 附带的展会。从 2008 年开始,为了促进图形学在亚洲的发展,又增加了 SIGGRAPH Asia 这个会议,各项组织与 SIGGRAPH 类似,每年不同时间举行。之后是在欧洲举行的 EuroGraphics,以欧洲各院校为主,但也吸引很多其他洲的学者参加。同时,SIGGRAPH 和 EuroGraphics 还举办了很多专项会议,包括 Symposium on Interactive 3D Graphics and Games (I3D),Eurographics Symposium on Rendering (EGSR),Non-Photorealistic Animation and Rendering (NPAR) 等,会议录取论文相对较少,但质量都比较高。在亚洲方面,还有 Pacific Graphics,

在 SIGGRAPH Asia 出现之前,是在亚洲地区最主要的会议。另外,一些实验室经常参加的会议还包括 Computer Animation and Social Agents (CASA), Computer Graphics International (CGI) 等。国内图形学方面的会议包括 China Graphics, IEEE International conference on CAD and Computer Graphics。

另外,我们的一些项目中,也会关注一些视觉方面的会议,这其中包括 IEEE International Conference on Computer Vision (ICCV), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), European Conference on Computer Vision (ECCV)。

在期刊方面,可视化最主要的期刊是 Transactions on Visualization and Computer Graphics (TVCG),除了一般性文章外,Vis 等会议的文章也会这上面发表。而图形学方向上,ACM Transactions on Graphics (TOG) 是最主要的期刊,类似的 SIGGRAPH 和 SIGGRAPH Asia 上的文章,会在 TOG 上发表。之后是 Computer Graphics Forum (CGF),EuroGraphics 和 EuroVis 的文章会在这个杂志上发表。在视觉方面,最主要的杂志是 IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)。

上述会议或期刊一般都有每年会议的官方网站,可以查询的每年录取的文章,如果进一步需要了解文章的内容,我们会在下一节中,介绍主要的搜索方式。

6.1.2 文献查询方法

随着网络的飞速发展,现在有很多途径可以搜索各种资源,在这里,只是简单介绍一些法,很多新兴的数据库,还有待我们的发现。

最近发展的比较完善的图形学论文查找的网站是 Kesen Huang 的个人主页,上面记录了近几年来上面提到的会议的所有论文,以会议做分类,可以很清晰的找到每年各个会议的文章。只提供作者给出的论文及项目连接,及各个作者主页链接,没有具体的文章。这个网站最早起源于 Tim Rowley 所发起的对 SIGGRAPH 论文的收集,方便人们查找,后来收集的会议越来越多,基本上覆盖了图形学界所有主要会议。很多人之前自己收集了一些会议的内容,也主动加入到这个网址中,为图形学及可视化方面论文的搜索提供了很大帮助。Kesen Huang 的网站只提供近年来的文章,但时效性强,每年会议刚刚录取的论文,只要作者给出文章链接,网站都会收录,甚至有些作者在文章放在网络上后,也会主动把信息提供给他们。

实验室也收集了很多的历年图形学方面的论文 (<http://www.cad.zju.>

edu.cn/internal), 其中 SIGGRAPH 的论文最全, 可以找到很多早期的文章, 另外, 一些在互联网上不易下载到的文章, 也可以在这里寻找。

接下来, 介绍一些常规的论文查询方式, 最主要的, 无疑是 Google, 以及它提供的 Google Scholar 搜索。Google Scholar 提供了强大的论文搜索功能, 几乎可以搜索到所有已发表论文, 及很多技术报告。但是这些链接并不一定可以下载到文章, 有些只是链接到各种论文数据库, 需要相应的数据库账号才可以下载, 各种数据库在之后会有简单介绍。同时, 利用 Google 可以查询到作者的相关信息, 找到一些论文之前和之后的相关工作。两个搜索各有优劣, 使用时可以两个都尝试一下。

可视化及图形学方面, 主要有两个组织, Association for Computing Machinery (ACM) 和 Institute of Electrical and Electronics Engineers (IEEE)。两个组织都提供了完整的数据库, ACM 的 portal.acm.org 给出了所有 ACM 的文章, 但只提供会员下载。IEEE 类似的数据库为 ieeexplore.ieee.org。我校是这两个数据库的会员, 所有浙大师生可以直接下载。但是, 实验室链接外网的线路是单独架设的, IP 地址段不属于学校, 而这些数据库的会员认证是根据 IP 的, 所以, 通过实验室的网络无法下载。如果需要下载, 可以通过学校的线路 (vpn 或者代理) 链接外网, 从而可以下载上述数据库的内容。

另外, 学校图书馆实际上有相当丰富的网络资源, 这其中包括各种学术组织, 期刊, 私人数据库的会员, 可以通过图书馆网站访问 (<http://libweb.zju.edu.cn>)。这其中包括大量的期刊, 及书籍, 期刊可以在外网的数据库中找到, 这大多需要 50 元 VPN, 有些有国内镜像的, 或者国内直连通道的, 可以用 10 元 VPN 访问。很多数据库也收录书籍, 可以通过网络浏览查看, 进行单页的打印, 但通常不允许下载或整本的打印。

另外, 还有一些关系数据库, 提供各个作者的文章列表, 合作者等信息。这方面最出名的是 DBLP, 它虽然没有提供任何文章信息, 但对作者搜索帮助很大。另外, CiteSeerX 提供对文章的引用信息, 标明是他人引用还是自己引用等, 提供部分文章的下载。

在寻找文章的同时, 还需要关注一下作者的个人主页。有很多东西, 无法在文章中体现, 主页上, 会有一些视频, 应用程序, 代码等。还可以了解到作者近期的工作进

展。有很多工作,基本上局限于几个人的工作,就可以了解全貌了。同时,有些作者还会对现有的工作做一些综述,方便新人学习入门。

6.2 常用网址

Kesen Huang 主页:<http://kesen.huang.googlepages.com/>

Tim Rowley 主页:<http://trowley.org/>

实验室论文收藏:<http://www.cad.zju.edu.cn/internal/>

Google:<http://www.google.com/>

Google Scholar:<http://scholar.google.com/>

ACM 数据库:<http://portal.acm.org/portal.cfm>

IEEE 数据库:<http://ieeexplore.ieee.org/>

学校图书馆:<http://libweb.zju.edu.cn/>

DBLP:<http://www.informatik.uni-trier.de/~ley/db/>

C-DBLP:<http://www.cdblp.cn/>

CiteSeerX:<http://citeseerx.ist.psu.edu/>

7 书籍和常用学习资料介绍

7.1 C 语言学习

"The C Programming Language", 经典中的经典, C 语言学习书籍中的圣经。推荐所有学习 C 语言的人看, 原因很简单, 本书的作者发明了 C 语言。如果你还没有仔细看过这本书, 找个时间好好洗礼一下吧。

"Lions' Commentary on Unix", 莱昂氏 UNIX 源代码分析, 该书的上半部分是完整的 UNIX 源代码(版本 6, 较早期的版本), 下半部分是莱昂氏对代码的详细分析。很多时候 UNIX 不仅仅只是一个系统, 从源代码中能够学到更多 C 语言的知识。

LINUX 源代码, 如果你想进一步提高你 C 语言的水平, 那么去看 LINUX 源代码吧。附带还能学习一下代码风格。<http://www.kernel.org/>

7.2 C++ 语言学习

"The C++ Programming Language", 由 C++ 之父: Bjarne Stroustrup 所写的书, 不过起点较高, 不太适合入门用, 但是如果你能把它看完, 一定会受益匪浅。推荐该书的读法, 不要追求一次都看懂, 隔一段时间再读, 会有新的体会。

"Programming: Principles and Practice Using C++", 同样由 Bjarne Stroustrup 编写, 2009 年底问世, 这次的目标读者是入门者。

"The C++ Standard Library: A Tutorial and Reference", 懂封装懂继承懂多态, 这样就算掌握 C++ 了么? 恐怕不行, 因为除了 OOP(面向对象编程), GP(泛型编程) 同样是 C++ 语言不可或缺甚至是比 OOP 更具活力的机制, STL(Standard Template Library, 标准模板库) 作为 GP 的优秀代表就是所有 C++ 程序员手中有力的武器, 这本书就是告诉你怎么使用这把武器的。

"STL 源码剖析", 知名华人 C++ 专家、台湾同胞侯捷先生的作品, 市面上难得的 C++ 中文原创佳作, 如果你已经掌握了 STL 这把武器的用法但并不满足, 还想拆开一探武器内部的究竟, 那么这本书你不能错过。

7.3 图形学基础学习

"Computer Graphics: Principles and Practice in C", 本书介绍了图形学基础算法, 包含了计算机图形学的方方面面, 如果你想从基础学起, 那么这本书不能不看。

"Real-Time Rendering", 实时绘制方面最好的教材, 该书按照图形绘制流水线的各个部分, 讲述了实时绘制中的经典算法。

"计算机真实感图形的算法基础", 真实感图形学的经典教材, 里面详细介绍了真实感绘制的经典算法, 包括光线跟踪和辐射度算法。(可能现在已经绝版)

7.4 OpenGL 学习

"OpenGL Programming Guide: The Official Guide to Learning OpenGL", 俗称红宝书, 学习 OpenGL 的入门书籍, 但是它不仅仅是入门书籍, 更是一本工作手册, 当你遇到 OpenGL 问题的时候, 翻一翻红宝书, 总会有惊喜。(本书的版本通常会随着 OpenGL 版本一起更新, 本指南编写时, 为第 7 版。)

"OpenGL SuperBible: Comprehensive Tutorial and Reference", 俗称蓝宝书, 红宝书更像是字典, 而该书以实例的方式讲述 OpenGL, 喜欢从例子入门的同学可以看这本书入门。

"OpenGL Shading Language", 橙宝书, GLSL 的入门教材(可能是唯一的 GLSL 教材)。

NeHe OpenGL Course, NeHe 的 openGL 教程, <http://nehe.gamedev.net/>, 共有 48 课, 每课讲述一个主题, 并且提供源代码, 是除了以上三本书之外最好的 OpenGL 教学资料。

7.5 Direct3D 学习

D3D 学习资料不少, 但是最好的一定是 DirectX 的帮助文档, 该文档随 DirectX SDK 一起提供, 里面包含了入门知识, 入门例程, 高级应用实例等等, 学习 Direct3D, 只需要查看帮助文档就可以了, 不再需要看任何其他书籍。包括 HLSL 都可以通过该文档学习。

7.6 CG 语言学习

学习 cg 语言不是必须的, GLSL 和 HLSL 都能代替它的作用, 但是学习 cg 语言之前必须学好 OpenGL 或者 Direct3D。

"The Cg Tutorial". Nvidia 官方的 cg 语言入门教材, 写的深入浅出, 适合初学者学习。现在免费能够在 Nvidia 网站上可以阅读。

7.7 GPU 编程学习

GPU 编程学习除了需要学习 GLSL, HLSL, CG 等 GPU 编程语言外, 还可以通过学习他人的例子来进一步优化自己的程序执行顺序, 数据结构, 或者学习一些最新的 GPU 编程技巧。

"GPU Gems I,II,III", 这一个系列三本书是讲述 GPU 编程技巧, 书中有大量的实例以及现成的代码, 通过对这些实例的学习和研究, 能够提高自己在 GPU 编程方面的技巧, 提高自己编写的 GPU 代码执行效率。

7.8 CUDA 编程学习

"CUDA Programming Guide", Nvidia 官方的 CUDA 学习资料, 包含于 CUDA Toolkit 内, 可以在 Nvidia 网站上下载到。

7.9 MFC 编程学习

"Programming Windows with MFC", 微软出版社出品, MFC 学习中的精品之作。看此书切忌浮躁, 多想多实践, MFC 其实不难学。

学习 MFC 时遇到不懂的时候, 可以使用 MSDN。

7.10 QT 编程学习

"C++ GUI Programming with Qt 4", 学习 QT 的好资料。

7.11 编程杂书推荐

"程序员修炼之道", 将宏观思想理念, 不讲编码细节。

" 算法导论", 系统学习算法的好书, 也是诸位将来应聘 IT 技术岗位时必备之宝典。

"Code Optimization: Effective Memory Usage", 代码优化中的代表之作, 值得一看。

"the Art of UNIX Programming", UNIX 的艺术可以让人一生受用, 这本书不止是讲述 UNIX 编程, 更是讲述 UNIX 文化, UNIX 之美。

" 深入理解计算机系统", 本书的副标题就是推荐的原因: 从程序员视角。

8 小丁讲编程

8.1 总论

从系统说起, windows vs linux (这里的 linux 你可以理解为 linux, 或者任何非 windows 系统), (如果读者是一个坚定的自由软件支持者, 或是坚定的鄙视使用盗版者, 那么恭喜你, 可以跳过这一大段, linux 是你最好的选择, 我相信你也完全有动力有能力去解决在 linux 下开发的一些问题, 事实上在 linux 下已经有很多成熟的图形或可视化开发实例) 选择哪一个系统作为开发平台, 这是一个经典的难题, 不过从我们特殊的应用角度出发 (图形图像, 可视化), 系统平台的选择可以有一些简单的结论:

- 首先, 最最重要的是看你手中项目的开发周期。如果你那个项目的开发周期只有一个月或者更短, 比如老板要你实现一个算法, 准备一个月后投稿, 那么不要想太多, 选择你最最熟悉的平台, 最最熟悉的环境; 而如果你的项目是一个长期的, 而且你也愿意去尝试一些新的事物, 学习一些新的东西, 那么可以选择你不擅长的开发平台。
- 其次, 在宏观上讲并不存在 linux 优于 windows 或者 windows 优于 linux 的事情, 所有的平台上都有人开发了很多成熟的图形或者可视化软件, 并且有些程序在多个平台上都有实现。需要注意的是, 对于一个图形或者可视化程序, 抛开显卡驱动的实现效率问题, 两种平台上的效率几乎是等同的, 决定程序效率最大的因素是算法本身, 其次是最于特定算法的特定优化 (包括优化数据结构, 程序执行顺序等), 再次才是平台差异性。
- 第三个结论是选择 linux 的一些特殊情况:
 - 如果你的项目要用到集群来加速计算, 那么请把你的核心代码在 linux 平台上实现, 现在实验室内大多数的集群上跑的系统是 linux。

- 如果你的项目需要更多的和系统的底层打交道,例如需要高速的 I/O,或者频繁的网络调用,那么可以考虑选择 linux,这些操作的开销 linux 往往会比 windows 来的小。
 - 如果你考虑到一些开源的问题,那么最好选择 linux 平台下做开发。
 - 如果你的项目是可视化方面的,你为了使你的项目看上去更加专业,也可以使用 linux,很多专业的可视化软件最初都是在 linux 平台下做的。
- 第四个结论是选择 windows 的一些特殊情况:
 - 如果你手中的显卡芯片是非 nvidia 公司生产的(主要包括 ATI /AMD, Intel),那么很不幸,你最好选择 windows 作为你的主要开发平台(为什么加“主要”二字?当你开发一些图形要求很小的程序,例如你不需要用到 OpenGL1.5/OpenGL2.0 扩展以上的情况,那么你还能够继续选择 linux 平台),因为在 linux 平台上 nvidia 公司的驱动支持是最完善的,也是和 windows 平台同步的,其他的公司显卡驱动对 linux 等非 windows 平台支持并不太完善,有很多 OpenGL 的高级扩展不能够正常使用。
 - 如果你对 DirectX 的熟悉程度大于 OpenGL,而且你在短期内不想尝试 OpenGL,那么你能只能选择 windows,原因很简单,linux 上没有你要的 DirectX(或者说原生的 DirectX)。这个结论可以做一个简单的推广,比如你对 MFC 或者 Windows API 很熟悉,比如你很喜欢 VS 这种可视化的编程方式(事实上在 linux 下也有开发软件采用和 VS 相似的风格)等等。
 - 如果你想开发一款游戏,或者类似游戏的一款图形软件,那么推荐使用 windows,在 windows 下可以使用 DirectX,非常适合游戏的开发。
 - 第五个结论是一些特殊情况(以后可以慢慢扩充):
 - 如果你的工作是在前人的基础上的延续,那么最好选择与前人相同的开发平台,除非你打算从零开始。

以上是总结出来的一些结论,但是要记住的是最终做决策的是项目本身的要求以及你自己的要求,不要被以上一些条条框框限制住。

8.2 Windows 下开发软件介绍

- 首先是最经典的 VS(Visual Studio)从 6.0 到 2010, Visual Studio 越做越强大, 功能也越来越强。简单的说安装好 VS 你不需要特别的设置就能开发你的程序, 从代码编辑, 编译调试, 到最后程序的发布, VS 都能够帮你搞定, VS 提供的是一站式服务。VS 包括 c, c++, c#, VB 等编程语言。对于 VS 版本的选择, 有如下一些建议(主要针对 C++ 编程):
 - 默认情况下, 请选择 VS2008, 别忘了打上 sp1 补丁;
 - 如果你对 C++0x 感兴趣, 可以尝试 VS2010;
 - 如果你想复古, 想要启动 IDE 就像启动记事本一样的速度, 并且你只用 C 语言, 你可以玩玩 VC6.0;
 - 没有特殊的情况, 尽量不要去选择 VS2005, 这一代的 VS 自身 bug 比较多;
 - 如果使用 VS2008, 推荐下载安装 Feature Pack, 具体的介绍可以在网上搜索关键字“Feature Pack”;

需要注意的是从 VS2005 开始, VS 分为收费和免费两个版本, 如果你觉得你永远都用不到 MFC(这里主要针对 c 和 c++ 开发), 那么你可以选择免费的 Express 版本。

8.3 VS 插件介绍

- Intel 出品的编译器 Intel C Plus Plus Compiler, 最强大的编译器, 能够提供最强大的编译器优化, 该编译器能够对你的代码做深度的, 针对特定平台的优化, 能够支持最新的 Intel 处理器特性(包括 SSE4), 需要注意的是编译过程相当缓慢。该编译器能够集成到 VS 中, 有图形化的配置工具, 使用起来相当方便。
- 然后是 Visual.Assist.X, VS 的插件, 能够更加方便的编辑你的代码。不过不是特别推荐这个软件, 使用这个软件越久, 要戒掉的概率越低。推荐在学习编程

的初期使用,到了中后期就不需要这个工具辅助你编程了,养好良好的编程习惯比依赖这个软件更加重要。

- Intel VTUNE, C 或者 C++ 程序性能剖析器,该软件能够帮你找到程序的热点(即当前程序中最消耗计算资源的代码片段),从而对热点区进行优化,提升程序性能。
- Java 开发工具 Eclipse,其实 Eclipse 已经超出了 Java 开发的范畴,但是在 windows 下有了 VS,因此它在 c 和 c++ 语言开发中的应用被无限得缩小了。如果你习惯了用 VS 开发 c, c++ 或者 c#,那么你应该很习惯用 Eclipse 开发 Java。
- GCC for Windows,如果你是从 Linux 下转过来的开发者,如果你一时间难以适应 VS 这种包办的风格,那么推荐使用 MinGW 和 Cygwin。MinGW 这个系列的移植,主要方向是让 GCC 的 Windows 移植版能使用 Win32API 来编程。而 Cygwin 的目标是能让 Unix-like 下的程序代码在 Windows 下直接被编译,它提供了 shell。
- Ultraedit,你可以把它当作一个编辑器来看待,是系统自带的记事本的绝佳替代品,并且是编辑网页的第一选择,作为一个 CS 出身的人,应该拒绝使用 Dreamwear 这种东西,表格才是 HTML 的精髓(至少这个是我个人的感觉)。
- Vim/Gvim, vim 在 windows 下的移植,同样适合从 linux 下转过来的开发者。

8.4 Linux 下开发软件介绍

Linux 下最经典的开发环境莫过于 vim/Emacs + GCC+ GDB。

- GCC, Linux 下编译器的第一选择, c++ 的版本是 G++。
- GDB, 感觉和 GCC 就是天生的一对,一个编译,一个调试,两者形影不离。
- Vim/Emacs, Linux 下运用最广的编辑器,确切地说 Emacs 已经超托了编辑器的范畴,在这里我们暂且把它当作编辑器吧。两者走得是不同的风格, vi/vim 是走简约风格, simpler is better, Unix 设计的宗旨之一;而 Emacs 复杂的多,你可

以在 Emacs 上干任何事,我看到过用 Emacs 下 BT 的家伙。(插一句,用 Vim 的需要注意你的 ESC 键,而用 Emacs 的要看好你的左手小拇指。)还有,这两者都是可以抛开鼠标的,你的双手只要放在键盘上就可以了。

- Linux 下也有一些类似 VS 的集成开发环境,例如 KDeveloper, Anjuta, 前者在 KDE 环境下,后者在 GNOME 环境下。还有 Eclipse, 在 Linux 下它也主要是用在 Java 编程上。
- Intel C Plus Plus Compiler, 很高兴,这个在 Linux 下也有对应的版本。只是这次它没有了图形化的配置工具,和大多数 Linux 中的程序一样,要在后面跟一堆参数进行设置。注意,该软件是收费软件。

编程语言的选择基本上只要遵循以下两条规则就可以了:

- 在大多数情况下我们推荐使用 C 或者 c++ 语言作为开发的主要编程语言。这一点基于两个考虑,首先图形 API 具有完整的 C 或者 C++ 接口,其次能够使得其他人更加方便的使用你的代码。在某些应用条件下可以使用 Java,如信息可视化(infovis)的研究开发,因为一些主流的用于 infovis 的库(如 prefuse、processing)是使用 Java 编写的。
- 不提倡完全使用汇编语言来进行开发,对于算法的核心部分,需要使用 CPU 一些特殊指令(如 Intel 的 SSE 指令)来提高效率时,鼓励使用嵌套汇编语言。

8.5 图形 API 介绍

Direct3D 是 DirectX 中的一个子集,主要负责图形绘制,注意 Direct3D 不仅仅能够应用在三维绘制上,在后期的 Direct3D 中集成了 DirectDraw,因此它是极好的二维矢量绘制库,由于它是可以直接调用显卡底层驱动,它对二维图像的绘制效率远远大于普通界面库提供的二维绘制接口。Direct3D 只有在 windows 平台上提供,在 Win7 上最新的版本是 Direct3D 11,在 XP 上最新版本是 9.0c。HLSL 是 Direct3D 对应的可编程着色语言。由于该图形库发展迅速,能够支持最新的显卡性能(往往新的 Direct3D 版本先于显卡发布),现在几乎所有的 Windows 平台上游戏都采用

Direct3D 作为底层的图形绘制 API。不过 Direct3D 起点比较高,入门需要较长的时间。

OpenGL OpenGL 最大的优势在于跨平台,在可视化领域广泛应用 OpenGL (一是因为很多可视化软件是跑在 Linux 平台上,二是可视化领域通常不需要用到太多最新的显卡技术)。OpenGL 的起点较低,在更多的时候它是作为一种教学用的图形 API 和工业图形 API 而存在的。GLSL 是 OpenGL 的可编程着色语言。OpenGL 和 Direct3D 先比较显得相对简单,它仅仅是图形绘制的 API,而 Direct3D 还包含了一些常用的图形计算数学库(有时,简单也是一种美)。近年来 OpenGL 的更新相对缓慢,往往要落后 Direct3D 半年以上。现在最新的标准是 OpenGL4.0。

Direct3D 和 OpenGL 的比较,作为新手入门,选择 OpenGL 比较好,它的起点较低,当你慢慢对图形有了感觉之后,可以选择学习 Direct3D。而在实际的项目中,如果是在 Linux 平台下,那么只能选择 OpenGL;如果有最新图形绘制 API 的支持,并且在 Windows 平台下,那么最好选择 Direct3D;除此以外,可以随意选择,不过多数情况下推荐用 OpenGL(使用起来相对简单)。

8.6 常用库介绍(C/C++)

8.6.1 数学库

Intel MKL 可能是世界上最快的在 CPU 上实现的数学库。功能强大,提供 C 和 Fortran 接口。该数学库包含了三层 BLAS 库,LAPACK 库,傅立叶变换(DFT 和 FFT),以及多种随机数生成接口等。需要注意的是,该数学库是收费软件。

Taucs 该数学库用于求解稀疏线性系统,提供 C 语言接口。是跨平台的数学库,开放源代码,在不同平台上可以自行编译。最后一次更新在 September 4, 2003: Version 2.2。

8.6.2 界面库

MFC Windows 平台上最常用的界面库,通常由 VS 自带。MFC 其实并不像有些人评论的那样不堪,虽然学习 MFC 是一个相对痛苦的过程,但是 MFC 的资料确实

最多的。在 VS2008 中可以安装 Feature Pack,使得用 MFC 也能画出 Office2007 般的界面风格。如果使用 MFC,推荐安装这个。

下面是三个著名的跨平台界面库的比较,摘自国际互联网:

- **GTK+.** GTK+ 主要用在 X Window 上,整个设计的架构和许多概念和 MFC 以及一般 Windows 上的程式开发大异其趣,入门门槛较高,而且最主要的特色是,他用不具有物件功能的纯"C"语言,模拟物件导向。所以写起来比较复杂艰涩,而且充满大量巨集,使用和除错都不是很容易,但优点则是可以用 C,不需 C++,如果和 Win32 SDK 比较,不会难学多少,缺点是不易上手使用,而且文件比较缺,架构又非常复杂,且提供的东西比起其他无所不包的 library,是简陋了一点,函数命名又臭又长。对于简单的程式,GTK+ 会显得太复杂,但是当你开始想扩充其他 library 也都没提供的进阶功能,就会开始赞叹 GTK+ 的架构严谨,还有超乎想像的高度弹性。同样的东西要用 MFC 来做反而会要人命,并且多国语言的支援良好,内部也全面使用 UTF-8,相容性好,又是 Unicode 能够习惯的话,GTK+ 值得推荐,但没有很建议学,毕竟不好学,要用到熟会需要比较久,而且那样很多 C++ 的功能会用不到。GTK+ 有 C++ 版本叫做 GTKmm,没用过但看文件觉得,并没有比 gtk+ 简单到哪里去。因为 gtk+ 本来就是物件导向,所以即使换了 c++ 语言,写起来架构还是差不多的。另外,gtk+ 有 Windows 版本,但缺点是,执行缓慢,不稳定,而且介面是使用 gtk+ 自己的,不是使用 Windows 内建的"Native"原生图形介面,看起来会不太习惯。Mac OS X 下可用 X11 来执行 gtk+ 但那样出来的程式是长得像 UNIX 程式,而不是美美的 OS X Aqua 外观。
- **QT.** Qt 的功能,应该是这三者加上 MFC 之中最强大的,文件也很完整,又有 RAD 工具可以辅助开发,并且有商业公司做强力后盾。不但有 Windows/X Window/Mac 版本,甚至还有嵌入式系统可用的版本,稳定性还不错,物件封装也算良好,资源比 GTK+ 或 wxWidgets 多得非常多,而且发行公司提供了相当多范例,算是一家以开放原始码成功营利的模范公司。知名的 KDE 整个是用他开发,证明了他的稳定性和强大功能。缺点是如果你用他开发非 GPL 开放程式码的软体,必须以极昂贵的金额,购买商业版本。而他的图形介面并不完

全是"Native GUI",只是透过 **theme** 去模拟系统上的标准 GUI,所以看起来很像,却会有些地方可以明显看出破绽。执行速度缓慢还有过於庞大则是另一个问题。虽然封装得很良好文件也齐全,并不代表他就很容易学还有一个严重问题是,他写的不是标准 C++,他使用的 **signal/slot** 机制必须透过 Qt 提供的 **preprocessor** 处理过才可以转送给编译器,这部份可能被限定用 **qmake**,算是一个可惜的地方,不过瑕不掩瑜,还是很推荐。忘了说,他内部也是 **unicode**,多国语言没问题。

- **wxWidgets**。**wxWidgets** 和 **MFC** 最接近,命名习惯或架构都高度相似,会 **MFC** 几乎不用重新学习有十余年历史,此外,他的物件封装比 **MFC** 要好,提供的功能也多上太多,又跨平台一般知名的 **MFC** 程式都会选择用 **wxWidgets** 改写,来快速移植原程式到其他平台例如 **eMule** 用 **wxWidgets** 移植出 **aMule**, **xMule**, 还在开发中的 **Filezilla 3** 等,而他最主要的特色是,他是"跨平台"的"Native" GUI toolkit,在各种平台上都可写出使用该平台内建 Native 原生图形界面的程式。在 **Windows** 上就长得跟其他 **Windows** 程式一样,在 **Linux** 下就使用 **gtk+** 的图形界面,在 **Mac OS X** 下就可以使用华丽的 **Aqua** 外观风格,这点是非常强悍。不像 **gtk+** 到其他系统都还是只能用 **gtk+**。自己的缺点是,中文支援在有些地方会出问题,例如剪贴簿的操作。得自己 **patch** 但仍然相当推荐,即使是个庞大的 library,效能依旧不会太差,尤其在 **Windows** 上执行速度并不输 **MFC**,与其学 **MFC**,不如学 **wxWidgets**。

一些轻量级的图形库,上面介绍的都是一些庞大的界面库,如果你不需要这么多的功能,只想画个界面,做几个按钮,那么你可以选择一些轻量级的界面库,方便不同用户使用你的程序。

- **glui**。是 **glut** 的扩展,提供了一些常用的控件,能够处理一些常见的用户交互。如果你使用 **glut**,那么你不应该错过 **glui**。
- **FLTK**。The Fast Light Tool Kit,一个轻量级的 GUI 开发库。但这轻量级并不代表功能的羸弱,相反,FLTK 在具有基本的 GUI 功能之外,还拥有一些特殊的功能,比如跨平台、内置 **OpenGL** 功能、速度更快、尺寸更小、协议宽松等。当然,

缺点也是有的,比如对于复杂的界面构件支持不够,资源支持的不足等。但一个工具如果使用的好,取其长而去其短,自然可以飞花摘叶皆可伤人。但是无论如何,FLTK 的目标还是针对嵌入式和封闭软件的开发,所以复杂的界面并非其长项,如果想做出花哨复杂的界面,还是用其他 GUI 库比较好,FLTK 并不适合。

8.6.3 计算机视觉库

- OpenCV。Intel 支持的一个开源视觉计算库,里面实现了许多经典的视觉算法,现在也包含了机器学习的功能,也有一个小型的界面库。该视觉库的功能不花哨,但都很实在。

8.7 小叶来补充(C/C++)

关于编程,真是说不完道不尽的,我就在丁师兄的基础上补充一些自认为有用的东西吧。

8.7.1 可读性 V.S 效率

可读性和效率这两者往往是矛盾的,哪个优先? 可读性! 一定是可读性! 因为代码不只是写给机器编译运行的,更是写给其他程序员看的。而且,牺牲了可读性就一定能带来效率上的提升么? 请不要低估现在的编译器水平,可读性高的代码同样可以经过编译器优化得到高效率的汇编代码。(当然,该考虑效率的地方还是要考虑的,比如非内置类型的参数传递优先选择传址的方式等等。)

还有一句名言与君共勉:“过早优化是万恶之源(premature optimization is the root of all evil)”——Donald Knuth。

8.7.2 美好愿景:小组编码规范

“这代码怎么写的啊? 搞不清楚他想干啥!”相信诸如此类的感慨出现不是一个小概率事件。承接上文,为了加强程序的可读性,提高组员之间程序思想沟通、项目移交等工作的效率,组内使用一个统一的编码规范势在必行。然而,并不是说只有标准是对的,不符合标准并不代表不对或者不好,但不符合标准就要像标准妥协,这对

于已经积累了相当代码经验的程序员来说的确是一件痛苦的事情。长痛不如短痛，还是向标准看齐吧。

不过目前小组的编码规范还没有出炉，即使出炉了也需要有人监督贯彻执行才有意义，这样的重任就交给本文的读者了，加油。

8.7.3 继续介绍库

一句名言：“不要重复发明轮子”。（题外话，从学习角度来讲，鼓励大家以优秀轮子为榜样多多发明自己的轮子。）首先，发明轮子需要制造轮子的时间，同时还要保证造出来的轮子是好用的、鲁棒的（怎么保证？用测试时间？用人格信誉？还是用rp？呵呵，这是个问题。）

如果已经存在了一些轮子满足你的功能需求，同时这个轮子也经过他人的保证是好用的，那么我们为何不拿来用呢？

什么？学习成本？是的，这确实是一个需要考虑的问题，不过放心，一个库如果得到广泛认可和应用，它的文档和示例肯定会满足令你轻松上手的。

- **Boost**。如果你觉得 C++ 标准库提供的功能还不够，那么请关注一下 **Boost**：<http://www.boost.org/>。Boost 也被誉为“准标准库”，许多 Boost 的开发人员来自 C++ 标准委员会，同时 Boost 中一些得到广泛认可的库已经进入了下一代 C++ 标准（如正则表达式、智能指针、哈希表、随机数生成器等等）。不过由于 Boost 涉及面太广，因此完全掌握 Boost 似乎不太可能，可以先通过文档简单了解 Boost 里面的各种库都可以做什么，等到有实际需求时，再深入学习进而使用。当然，如果你是一个 C++ **geek**，相信谁也拦不住你去品尝 Boost 源码的:-)
- **TBB**。Threading Building Blocks, Intel 的开源并行库。使用 TBB，用户不必关注线程，而是更多地去专注于任务本身。也许你要问 TBB 和 OpenMP 有啥关系？其实后者从严格意义上来讲算不上库，只是一个语言的扩展，从比较具体（线程）的角度解决并行问题，而 TBB 则试图让用户直接使用其封装好的并行容器、并行算法等等并行概念的事物，从而完成并行程序的构建。
- **Thrust**。基于 CUDA 的一个类 STL 库（当然，目前还没做到 100%STL 的功能），

简单的说,我们怎么用 STL 就怎么用 Thrust,只不过代码文件从.c++ 变为.cu,编译器从 MSVC/g++ 变为 nvcc,背后算法的实现都是通过 GPU 来完成的,这样的免费午餐,对于接触过 CUDA 编程的人来说怎能错过。

- TinyXML。XML 语言作为 W3C 标准被广泛应用于描述各类信息数据,如果你使用 XML 和他人沟通而对方表示看不懂,那么你一点责任都没有:-) TinyXML 就是一个轻巧的解析 XML 格式文件(或字符流)的库。
- freeglut。如果你喜欢 glut 的简洁,但是不爽它的陈旧(比如不支持鼠标滚轮)和闭源,可以试试用 freeglut 做替代品。

9 图形学需要的主要数学知识及资料

9.1 所需要的数学知识

在计算机应用技术中,图形学和可视化是需要数学知识相对较多的应用学科。对于图形学来说,只要微积分和线性代数基础扎实就可以了,对于不同的研究方向,还需要一些数值分析、微分几何和泛函分析的最基础的知识。数值分析涉及的范围太广,图形学基本上用到的是解线性方程组,这方面有现成的函数库可以使用,而且也有 GPU 上的实现,这里就不一一展开了。关于数学知识(也适用于其它知识),记住下面的原则:

- 不需要都学会了再去做问题,研究过程中遇到什么数学知识再去学相关的知识,掌握起来很快的。
- 公式都是纸老虎,公式背后想表达的概念,或者说“图形”,才是重点。

可视化研究主要需要数据处理相关的数学知识,需要从数据中发掘出难以看到规律,然后以图形手段表达出来,主要用到概率统计等相关知识,这个领域的知识很广,在研究中用到什么再去学吧。

9.2 搜集的资料说明

我搜集了一些非常基本的资料,我认为掌握这些知识,可以对今后的学生产生积极的作用,刚入学的研究生应该可以没有障碍的阅读并掌握它们,主要包括:

- 曲线和曲面的数学知识,主要包括课件和一些电子文档,在文件夹"曲线曲面"里。
- 一篇短文"Hidden Markov Models",介绍隐马尔科夫模型的知识。这是一篇很好的介绍性文章,通过一个很著名的模型来展示怎么从数据中挖掘信息,作出判断。

- 另一篇非常精彩的短文"Information Theory Primer", 介绍信息论的基本概念——熵, 读过之后你可以自己“定义”如何衡量一个数据包含了多少信息量。

10 微分几何简介

本章将延续上一章的风格,以对话方式介绍微分几何,希望能够在1-2个小时内让大家了解图形学中用到的微分几何的最核心的知识。对于图形学研究来说,懂得一些基本的曲线论和曲面论就足够了,更深入的知识,可以在做研究的时候去学习。

10.1 曲线的曲率

先说曲线论,空间曲线 X 是一个函数:

$$X : \mathbb{R} \rightarrow \mathbb{R}^3$$

对于 X 来说,我们常听到曲率,那么什么是曲率呢? 常见的课本用切线的微分来给出曲率的定义,这种方式几何意义不明显,我将根据"A Comprehensive Introduction to Differential Geometry" 这部大部头里的讲法来介绍。这部书块头很大,五本,每本都堪称砖头,我只读过前两本,第一本讲流形的基础知识,第二本正儿八经的开始讲几何。这套书的作者是陈省身的高徒,后来不玩数学去搞出版生意了,现在应该很富裕。曾经有人拿他这套书来采访他,他自己说不敢想象自己年轻时写了这么一套书。其实我觉得挺可惜的,他是个非常有才华的人,坚持搞数学的话一定能做出很好的工作。这套书的写法与传统课本非常不同,它按照微分几何的发展历史来展开介绍,显得非常有血有肉,读的时候很有那种“follow the master”的感觉,好像高斯、黎曼就在黑板上给人讲课一样,很是让人激动,反正我读的时候经常会忍不住想起来跳一下。我们现在面对的大多数课本里介绍的概念,都是经过高度抽象和梳理过的,看起来总让人觉得敬畏,让人有在庙里拜佛的感觉。这套书的讲法则让人觉得亲切很多,不仅知其然,而且知其所以然,要是多一些书是这样写的就好了。

下面我们把自己当成数学家,来探索什么是曲线的曲率。所谓曲率,字面上理解就是度量一条曲线上每一点处弯曲程度的量,对于一般的曲线,很难一下子给出定义,那么我们就先考虑简单的情况,这种思考方式是数学里最常用的。什么是最简单

的曲线？直线。显然，在我们的直观印象中，直线不是弯曲的，所以它每一点的曲率都应该是 0。接下来考虑复杂一点的曲线，比直线复杂，但是又最常见的曲线应该就是圆了。圆是最具对称性的图形，它上面的每一点都具有相似的几何性质，因此在定义圆的曲率时，我们可以认为其上每一点曲率是相等的，那么该如何定义圆的曲率呢？我们看图 10.1：

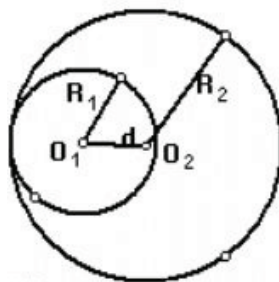


图 10.1 两相切圆

注意两圆的相切处，显然直观的看起来，在相切的地方，小圆的弯曲程度大于大圆的弯曲程度。什么量最能区别两个圆？半径。半径越大，弯曲程度越小，这符合我们对圆的曲率的直观印象。因此我们定义圆上每一点的曲率为这个圆的半径 r 的倒数 $\frac{1}{r}$ （定义越简单越好，其它关于半径的减函数也行，但是这个最简单）。当一个圆的半径趋于无穷大的时候，这个圆退化成直线，它的曲率 $\frac{1}{r}$ 为 0，这符合我们刚才对直线曲率的理解。接下来就顺理成章了，对于任意曲线 X 上的每一点 $X(t)$ ，我们计算在这个点处和这条曲线相切的圆（图 10.2），相切表示这两条曲线在这一点完美接



图 10.2 与曲线密切的圆

触，可以认为具有相同的弯曲程度，所以我们取这个相切圆半径的倒数作为 $X(t)$ 处的曲率。建立了这些概念后，公式的推导就只是例行公事了：首先计算 $X(t)$ 处的切线，然后在切线的法向上设未知点，建立圆方程，等等过程，解方程就一切 ok。如果

懒得推导(或者计算功底稍弱),可以翻书找公式,我也提供了相关的课件和资料,详细过程这里就不写了。需要注意的另一个量是挠率,它的几何意义很明显:偏离切面的度量,资料里也有。

10.2 曲面的曲率

10.2.1 Gauss 之前的工作

接下来介绍曲面的曲率,我们先看看在伟大的 Gauss 之前的数学家是如何研究曲面的。在 Gauss 之前,数学家们把曲面当做一件“衣服”,是由“线”织起来的,如图10.3所示:对于曲面 S 上的每一个点 p ,我们可以计算这一点的法向,经过这个法

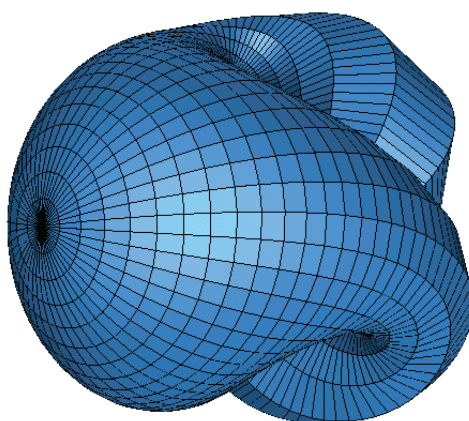


图 10.3 法平面与曲面相交的曲线

向有无数个平面,它们分别与 S 相交得到曲面上的不同的曲线,那么整张曲面可以当做是有一条一条的曲线“缝合”而成。既然有了曲线,我们就可以根据刚才的介绍计算它们的曲率。在这些众多的曲线中,有两条曲线 l_1 和 l_2 很特殊(对于一般“好”的,即非奇异的点都有这样的两条曲线),如果记它们的曲率分别为 k_1 和 k_2 的话(称为主曲率),那么其它所有曲线的曲率,都在 k_1 和 k_2 之间,并且,这两条曲线在 p 处的切向量,是相互正交的。这个就是数学神奇的地方,看起来毫无意义的乱七八糟的一堆曲线,它们之间竟然存在这种关系,而且,更加神奇的是(伟大的欧拉)下面的定理(图10.4):

定理 10.1: 对于另一条过 p 点由法平面切出来的曲线 l , 记它与 l_1 在 p 处的切向量夹角为 θ , 设 l 在 p 处的曲率为 k , 那么有如下的公式:

$$k = k_1 \cos^2 \theta + k_2 \sin^2 \theta$$

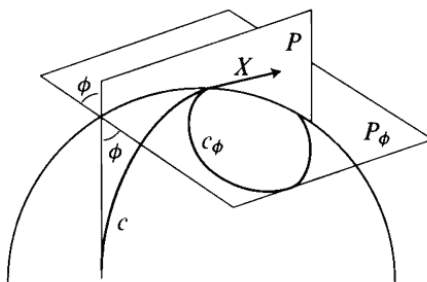


图 10.4 Euler 定理

除了这个神奇的公式, 还有别的也很神奇的定理如 **Meusnier** 定理, 具体可以查阅微分几何的书籍。一句话说来, **Gauss** 之前的数学家, 通过把曲面视为一簇曲线来进行研究。

10.2.2 Gauss 的工作

Gauss 是数学史上不世出的天才, 对于曲面论, 他的观点是: 曲面就是曲面, 把曲面视为曲线的集合固然能得到一系列漂亮的结果, 但是通过这种观点来研究曲面是远远不够的。 **Gauss** 的工作是革命性的工作, 揭开了微分几何的新篇章, 下面介绍他的工作。同曲面一样, 曲面论的核心问题也是曲率, 那么什么是曲面的曲率? 对于一个曲面 S 上的点 p , 如图10.5所示, **Gauss** 计算出 p 处的单位法向量, 然后把这个法向量平移到原点(称为 **Gauss Map**)。可以知道, 如果我们把 p 邻近的点都收集起来并且都做 **Gauss** 映射, 就得到一系列单位向量, 它们的端点构成了单位球上的一块子曲面。可以想象得到, 如果 S 在 p 处扭曲得越厉害, 那么由 **Gauss** 映射得到的子曲面的面积就越大(图10.6), 通过对上述过程取极限, 可以得到一个曲面的曲率的定义即

$$K(p) = \lim_{A \rightarrow p} \frac{\text{area } v(A)}{\text{area } A}$$

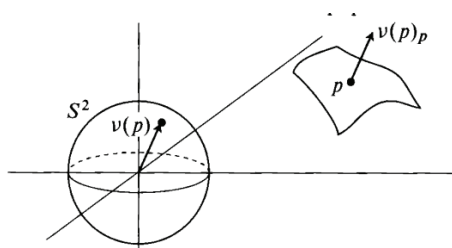


图 10.5 Gauss 映射

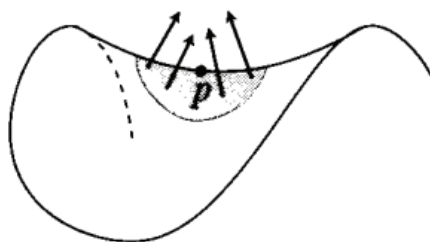


图 10.6 通过 Gauss 映射定义曲率

需要指出的是,这只是曲率的一种直观的几何解释,并不是严格的数学定义,比如如何取极限?不过这个缺陷并不致命,可以通过发展相关概念(微分形式)进行弥补。与严格的定义相比,两者的几何意义其实是一样的,区别在于后者更加形式化,更加严格。

在进一步介绍前,先看几个例子,来巩固,同时也是颠覆我们对曲面曲率的想法。首先是平面的曲率,如图10.7,显然对平面上每一点的法向量做 Gauss 映射后,都映

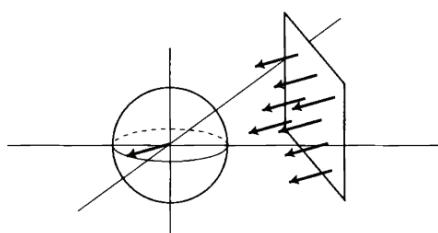


图 10.7 平面的 Gauss 映射

射到唯一一个单位向量,因此任何一点 p 附近的部分经过映射后,面积为 0。根据上

面的公式,平面曲率为

$$K(p) = \lim_{A \rightarrow p} \frac{\text{area } v(A)}{\text{area } A} = \lim_{A \rightarrow p} \frac{0}{\text{area } A} = 0$$

这符合我们对曲率的直观理解。

对于半径为 r 的球面,通过其 Gauss 映射(图10.8)不难看出其曲率为

$$K(p) = \lim_{A \rightarrow p} \frac{\text{area } v(A)}{\text{area } A} = \lim_{A \rightarrow p} \frac{\frac{1}{r^2} \cdot \text{area } A}{\text{area } A} = \frac{1}{r^2}$$

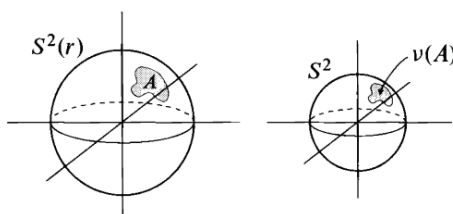


图 10.8 球面的 Gauss 映射

接下来的例子,会颠覆我们对曲率的理解。按照字面理解,弯曲的曲面,曲率必然不会为 0 的,但是把刚才的推导用于柱面时,我们可以看到,对于处在与 p 点同一高度上的那些点,它们经过 Gauss 映射后为单位球上的一个大圆(图10.9),我们知道大圆的面积为 0,所以柱面的曲率为

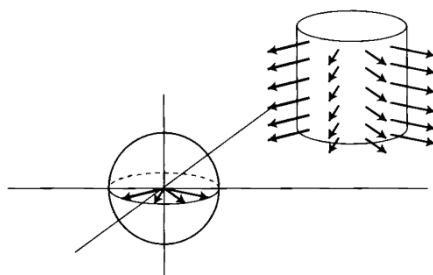


图 10.9 柱面的 Gauss 映射

$$K(p) = \lim_{A \rightarrow p} \frac{\text{area } v(A)}{\text{area } A} = \lim_{A \rightarrow p} \frac{0}{\text{area } A} = 0$$

这个看似诡异的现象,其实很好解释,这是因为,如果有一把剪刀沿垂直方向剪的话,柱面是可以剪成一个平面的,也就是说柱面可以展开成一个平面,而我们知道平面的曲率为 0,所以,柱面的曲率为 0。同样的,我们也可以通过类似的计算,得到圆锥面的曲率为 0 的结论(图10.10)。

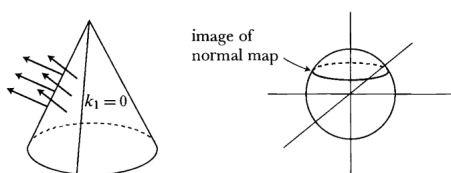


图 10.10 圆锥面的 Gauss 映射

接下来谈一下 Gauss 定义的曲率和他之前的数学家的研究之间的联系。在上一小节中,我们知道在曲面上任何一点处,可以得到两个特殊的曲线的曲率 k_1 和 k_2 ,经过推导可以知道,它们和 Gauss 定义的曲率之间有如下关系,即

$$K(p) = k_1 \cdot k_2$$

根据这个公式,我们也可以从另一方面解释刚才的几个例子。对于球,它的 k_1 和 k_2 都是一样的,为 $\frac{1}{r}$,因此球面的曲率为 $\frac{1}{r^2}$;对于柱面或者圆锥面,由于它们各自有一个主曲率为 0,因此它们的曲率为 0(图10.11)。从这些联系中,可以看到 Gauss 的天才

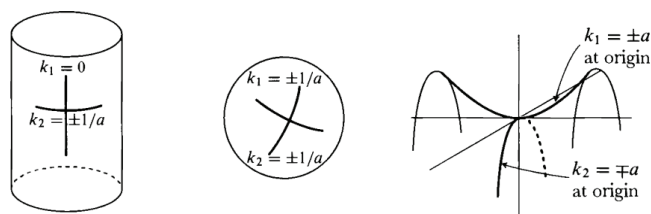


图 10.11 Gauss 曲率与主曲率的关系

之处。Riemannian 更是进一步的把曲率的研究推广到了高维,建立了流形概念的雏形和黎曼几何的架构,可惜他英年早逝,让人扼腕叹息造化不公。这两位不仅在几何上是达摩老祖一样的人物,在数学的其它领域同样是登峰造极,按《倚天屠龙记》来算的话,Gauss 和 Riemannian 都是数学史上张三丰似的人物,接下来的大数学家希尔

伯特,大概就差不多属于次一级的高手的功力了。所以强烈建议大家有兴趣的话,去读一读刚才提到的五卷书里的第二卷(可以不读第一卷的),在那里,你能见识到数学史上最顶尖的两个泰山北斗的工作。

当然,Gauss 的工作远不止这些,但是建立了这些概念后,只要不要被繁琐的公式牵着鼻子,多关注公式想要表达的几何意义,就可以没什么困难的阅读微分几何的书籍了。在这里给大家提个醒,有一个定理叫做 Gauss 的绝妙定理 (Theorema Egregium,没写错,这是法语)。它是说,我们按照刚才定义的曲率,其实是把曲面放在三维空间里计算得到的,即曲率是外蕴量。但如果按照 Gauss 的构想,只考虑曲面本身而不考虑它所在的三维空间的话,实际上曲率可以通过曲面本身的度量 ---通俗的说就是如何定义曲面上一条曲线的长度 ---来得到。这样,曲率就是曲面本身的内蕴量。这个定理需要慢慢体会,这是一个让数学家们感到震惊的结论。

最后提一下非常著名的 Gauss-Bonnet 定理,陈省身对这个定理的证明做出了杰出的贡献,开辟了微分几何的新篇章,奠定了他顶级数学家的地位(截止至 2009 年,他是唯一一个获得国际数学界最高荣誉“沃尔夫奖”的华人)。这个定理的一般形式是针对于 \mathbb{R}^{n+1} (n 是偶数) 的 n 维子流形的,对于二维有定向(莫比乌斯带没有定向)的封闭曲面的特殊情形,有下面的公式:

$$\iint_M K dA = 4\pi(1 - g) = 2\pi\chi(M)$$

其中 $\pi\chi(M)$ 是曲面 M 的欧拉示性数, g 表示 M 的亏格(即孔的个数)。这个公式是说,如果把曲面每一点的高斯曲率对这个曲面做积分,那么得到的结果是一个定值,这个值只依赖于这个曲面的拓扑,而与这个曲面的形状无关。也就是说,只要两个曲面有着相同的拓扑分类,比如说它们没有孔(球面,椭球面),那么上面的积分等于 4π ;如果只有一个孔(轮胎,面包圈),那么这个积分等于 0,以此类推。这是个非常非常漂亮的定理,等式两边实际上是两个完全不同的对象,一边是微分几何,曲率表示的是曲面的局部性质,跟曲面的形状紧密相关;另一边是拓扑学,表示的是曲面的整体性质。Gauss-Bonnet 定理将两个表面上没有联系的对象紧密联系在一起,是数学中的珍品。对于欧拉示性数 $\chi(M)$ 再多说两句,这是表示曲面拓扑性质的另一个量,从上面的等式,我们可以看出来 $\chi(M) = 2(1 - g)$ 。对于离散的情况,

$\chi(M) = v - e + f$, 其中 v, e, f 分别表示曲面网格的顶点数、边数和面数。对于没有孔的封闭网格, 可以看出来 $v - e + f = 2$, 这就是著名的欧拉公式, 大家可以自己验证一下四面体、正方体或者其它的封闭网格模型。这个公式有许多有趣的、别出心裁的证明, 非常美妙。关于 $\chi(M)$, 还有另一套非常漂亮的理论即 **Morse** 理论来刻画它, 这里就不多说了, 有兴趣的话自己去找来看, 目前做数据分析有很多工作是以这套理论为依据的。对拓扑学感兴趣的, 建议去上一下数学系黄兆镇老师的《点集拓扑》和《代数拓扑》, 都是非常好的课。

11 科技论文写作杂谈

11.1 论文写作注意事项

科技论文写作远比其他的英文写作简单、规则。我们这里指代的论文,是指 **peer-reviewed** 的,即由审稿人专门审稿同意发表的论文。发表 **peer review** 的学术论文,如果了解审稿人在审稿过程中的侧重点,那么对我们自己文章的写作无疑是有帮助的。

在我审核过的文章中,仅仅有非常少量的文章我是认识文章作者的,不得不承认,如果我与文章的某个作者相识,那么我了解他们以前的工作,因此对他们的工作比较信任。如果文章中有比较有名望的教授共同署名,文章的被接受的机会肯定会相对来说更大一些,特别是论文处于 **borderline** 的状态。一般说来,审核文章最主要的标准有三个。

- 创新性 (**originality**)。

编辑或主审人在找审稿人的时候,都是需要自己先浏览一下文章,然后找本领域比较熟悉、且在领域内有点学术信誉 (**academic credit**) 的人审稿,这样就可以一定程度上保证对文章创新性的判断。越是名气大影响因子高的期刊越是强调文章的创新性这一点。影响因子很高的期刊由于所收稿件比较多,编辑为了提高审核效率会第一时间直接拒绝那些明显质量不高,创新性不够的文章,而根本就不会把文章发给审稿人,省去了费力找合适审稿人的麻烦。比若说,如果你的论文是有关实验的,那么你的文章中是否介绍了新的实验方法? 你的文章是否发现了新的现象? 还是更多重复或者遵从 (**follow**) 前人的成果,仅仅对试验步骤作了些简单的修补? 如果是后者的话,而编辑恰好又是同领域的专家,他们就很容易判断出来你的工作的创新性的,那么你的文章被直接据掉也就不奇怪。

具体到我们的研究或者文章上,为了保证文章的创新性,我们做研究前一定要多做做学术文献检索工作 (**literature search**),一定要搞清楚同领域内有哪些问

题已经被解决,哪些问题有待解决;一定要找出那些比较重要的文献细读精读;一定要对领域内的专家最新动向有所了解.而体现在写文章上就是,在写摘要 (abstract) 和介绍部分 (introduction) 的时候,索引要写得清楚,全面。千万别小看文章的 introduction,内行看门道,如果文章 introduction 写得好,审稿人就马上能够判断出来你对领域的熟悉程度,他们对你的研究工作就会有一个比较好的第一印象,审稿人的审核情绪也会有所提高,自然文章顺利通过审核的可能性也就大。

在任何可能情况下,我的文章寄出去前,总喜欢找一位领域内名气大的学术水平高的老教授帮我修改 introduction 部分。特别是一般自己觉得比较不错的研究文章,在寄给杂志之前,让同领域的“高人”帮你把把关不失为对自己负责对研究负责的好办法。

文章的创新性 (originality) 当然还体现在文章主体关于研究方法的介绍,不过这里暂时不特别强调这一点。如果你觉得自己研究工作的创新性不够,那么建议不要直接往顶级刊物上投。要知道投稿也是有学问的,因为如果你经常往某个杂志投稿结果经常被拒,你在那个杂志编辑心目中的学术信誉 (academic credit) 就会很快丢失了. 这样你下一次如果再想往同样刊物上发文章就更难了。相反你的文章被某个杂志发表的比较多,那么你也会慢慢也会积累起来你的信誉,有了信誉没准那一天你会被期刊编辑邀请做期刊审稿人了呢. 所以我个人主张发文章从选择合适的期刊开始就要慎重。同样的道理,被杂志选为审稿人也是一样的道理,所以不要总听那些抱怨牢骚的话,不要文章一被拒就怀疑审稿人故意刁难,这种情况有但并不是主流,因为每个审稿人也需要考虑自己的信誉. 况且每篇文章不是一名审稿人说了算的,某位审稿人如果胡乱评文章,那几次以后他不仅没有机会再做审稿人,很可能他辛辛苦苦积累下来的学术信誉也会丧失的。

- 全面性 (completeness)

如果文章有一定的创新性,那么第二个重要的判断指标就是文章的全面性 (completeness)。这一点比较 tricky。我审核过的文章,还有我自己发表的文章

被审核,reviewer 多数是在此方面“做文章”。再严谨的研究者也可能会遗漏些自己没考虑到的地方,这时候,reviewer 会指出研究的不足之处,给作者建议让他补充新数据或者对某些结论增加些解释等等,最终 reviewer 给文章的评语多半是 **major modification**,这种情况对试验和计算方面的研究更常见。也有些情况,文章不足的地方太多,甚至由于不够严谨导致结论“错误”,这时候文章多半要被拒。对于很多投往期刊的论文,通常只要完备性足够了,reviewers 也没有特别的理由挑刺来拒绝了。

在从期刊编辑那里收到所投文章的 reviewer comments 的时候,一定要仔细查验那些建议或者意见。也确实会有些审稿人不负责任,甚至有可能他们都没有读明白你的文章,给出了比较荒唐 (**ridiculous**) 的评语或者建议,但不管怎样,还是应该多在自己身上找毛病。仔细想想是不是自己文章本身实在是不够严谨,甚至研究中有太多的不确定因素 (**uncertainty**)? 在修改文章时候,一定要认真考虑审稿人的建议并补充新的证明材料。如果文章被拒不要抱怨审稿人吹毛求疵,以我的经验审稿人不大有可能是因为作者名气不足而故意刁难作者。但是倒是有可能审稿人对于他们自己熟悉的作者的文章尽量多挑毛病,避免直接拒掉,这可能是那些有名望的教授发文章过程中能够占到的“小便宜”。

但是另一方面,大多数有名望的教授往往更看重自己的名誉,至少我认识的几位著名教授,只要文章中有他的名字,在文章投出去之前,他们都要自己最后亲自审核文章的。他们的审核甚至比通常的审稿人更要严格,而他们修改过的文章,不得不承认文章确实水平高,那么如果你觉得自己的研究确实不错,如果能找有名望的教授愿意帮你把把关是保证你文章质量的不错的办法。

- 写作技巧 (**writing**)

科技论文的写作技巧并不像小说那样要求高。但好的科技论文是可读性很强的。文章写作技巧对文章的发表与否,和文章发表后的引用情况都有影响。在文章审核阶段,审稿人在审核文章的时候,如果作者经常出现语法错误,那么几乎是他们最难以忍受的,这表明作者对文章的写作态度就不太端正。给审稿人的印象会非常差,这种情况下审稿人会对文章的审核更吹毛求疵一些也就不足

为奇了。另外,写得比较好的文章,可读性强的文章,在文章发表后能够引起更多人的注意,相对来说影响力 (impact) 也会更大,文章被索引次数当然就会更高。所以在发表文章前最好多修改几遍文章,哪怕是找别人帮助看看语法对自己文章质量的提高也是有帮助的。

从个人经验来说,我在写文章时,总是先写背景工作,引起工作动机,最后才写介绍部分。目前,在大家都倾向于 overclaim 自己的工作的背景下,一种非常简单有效地写法是,用 bullets 的写法,来一条一条地列出文章的贡献。比如:

This paper presents our efforts on coherent detail control of line drawings with the following contributions:

- A new image-space structure, called hierarchical line bundle tree, that enables adaptive detail selection and dynamic merging of line drawings with a computational expense bounded by the image resolution;
- A new proximity measure for lines that combines the geometric properties and conceivability of lines;
- An efficient density control algorithm that preserves temporal coherence and smooth transitions among different scales, and extensively considers the conceivability of lines, the density of lines, and the user focus for conceivably effective cues; and
- A multi-core CPU implementation that achieves interactive performance for large-scaled scenes.

在写相关工作时,通常用平实的语言描述前人工作。对于本领域比较活跃的课题组或研究人员,尽量都引用一两篇他们的工作,这个叫拜山头。对于每篇相关工作,尽量用正面的评价,以示尊重。在每一类相关工作介绍结束时,可以用一两句话比较相关工作和本文工作的差别,当然,通常只写本文工作的改进之处。

对于算法类文章,可以再第三节给出关键的算法或流程,使得 reviewers 可以直接选择感兴趣的子章节阅读。

在实验部分,通常需要考虑效率或质量的改进,应该有翔实的实验数据作为算法的支撑,同时,有和前人工作的对比,才具有说服力。当实验部分存在部分不足之处时,可以将这些不足之处写在未来工作里。

最后,希望大家要正确对待论文被拒。在我个人发表的文章中,就曾经有一篇被拒过。我为了那篇文章前后狠下功夫地做了一年左右的研究工作,当时被拒觉得很 unfair,但是过后想想却觉得“因祸得福”。我当时是这样做的,那篇文章被拒后我等了半年多才第二次发表出去,期间又进行了大量进一步的研究,并对原文进行了大量的修改。态度端正以后,会把坏事变成好事的。

11.2 导师在科技论文写作中的职责

通常,导师是论文的通信作者或责任作者。通信作者是生物、化学等看重知识产权的研究领域的产物。为了避免可能的争端,有些杂志甚至要列出每个作者的具体贡献。计算机学科大部分不注明通信作者,但国内高校引导性地要求职称评审时只统计通信作者和第一作者,导致出现了很多通信作者的论文。责任作者我的理解是,课题组主要完成地的责任导师(级别最高的那位)。这个隐式的看法导致很多著名的教授在手下的学生瞒着做事出现问题的时候感觉到自己很冤枉---事实上,这些问题出在课题组的管理上。课题组的导师应该对每位新来的学生显式地、严肃地进行始业教育,防弊这种事情的发生。并且要求所有论文在写作后交由导师审定,发表后交与导师再次勘察。在整个课题组要倡导一种正确的研究气氛和研究看法,课题组的整个水平才能有所提高。

浙江大学的何赛灵教授回忆,导师对学生非常认真、负责。“就拿对待学生的论文来说,每一篇他都要亲自动手修改三四遍才允许投稿。他在修改学生文章时,总是让学生坐在他的旁边指导修改。然后,他一边修改,一边告诉学生这里为什么这么修改。有十几篇文章经导师这样修改下来后,学生也能独立写出比较漂亮的英文文章了。”他说:“我的导师认为,文章一旦拿出去发表了,就要经得起科学的检验、经得起时间和世人的检验,一定要严肃、认真对待。”而何赛灵现在沿袭的,就是导师那种严谨的治学态度。

现在,何赛灵要求自己的学生不要急于发表文章,重要的是首先要夯实学科基

础。因此,他经常跟学生讲,做学术要有三“心”:静心、自信心和事业心。他说,从事科研要能够静心,追求目标要有自信心,展望未来要有事业心,要想成为一个真正的科学家,这三“心”缺一不可。

11.3 关于选题和执行力

- “要学会集中精力阅读,集中精力思考。一开始,你可能只能让精神高度集中几分钟,然后就开小差,就开始犯困。但是,要坚持锻炼自己的思维,逐渐的让自己能集中精力十几分钟,半个小时,一个小时……这跟锻炼一样,你开始练长跑的时候,可能跑个几十米就累得气喘吁吁,但要坚持每天锻炼,逐渐的,你就能跑几百米,几千米。一个人的研究能力取决于思维深度,更取决于在这个深度下持续思考的时间。”
- “阅读论文要坚持每一天都读,而非每个礼拜要来见我之前才来拼命读一个晚上。我们要在 SIGGRAPH 上发表论文,就如同参加这个领域的奥运会。你能想象一个球员每个礼拜只练一次,就去参加奥运会么?”
- “集中精力,是指阅读或思考专业性很强的东西。这是个很耗脑子的事情,就如同跑步——这跟读小说之类的闲书不同,我们能坐着读一整天闲书而一点都不累,因为那个就像慢慢散步一样。”
- “要有进取心。我知道你刚开始进入这个领域,很多都不明白,没有方向。但面对这种情况,有两种面对的方法:坏学生会就此沉沦下去,什么事情都不做;好学生会不断的去尝试,没有方向,那就尝试着推开每个方向的门瞧一瞧。但是,坏学生也可以改变。”
- “读研跟本科不一样,不是只要把课上好就可以的。判断一个研究生可以毕业的标准是,你在导师面前滔滔不绝的说出你的想法,你某方面的知识比导师多。”
- “你应当养成记录自己每天做了什么的习惯,以此督促自己不荒废时间。”
- “你必须学会改变自己,必须!”

- “我最觉得自己知识最多的时候是本科。那时候感觉什么都懂,什么微积分啦,傅立叶变换啦,电路啦,什么都那么容易,一学就会,一下子懂了这么多东西。但到了研究生阶段,发现自己什么都不懂:一看论文,这个名词不懂,那个公式不明白。突然发现自己可能只知道1么多东西要学,真的让人感到很害怕。你们现在一定也会这样,这是正常的,都会有这样一个艰难的过程的。”
- “研究生跟本科生完全不一样:本科生只要把课程好好搞懂就行了,很容易就能拿个 A;研究生就得更注重研究,要通过读论文来自学。原先通过课堂掌握知识的效率实际上很低的,可能老师光讲了一点东西,你就明白了,然后几分钟老师仔细讲解的时候你就可以开开小差,过一会儿再回过神来,听听新的东西,又明白了,再继续开小差……自学的话,效率很高的。”
- “我同学拿到普度大学的岗位,但他选择了 UNM。因为普度那里图形学组比较没落,几年都没出过多少 SIGGRAPH 论文。他宁愿来这里,从零开始,和实验室另一个 CS 新来的教授一道,开始的时候每年两三篇 SIGGRAPH 论文,然后逐渐进步,几年时间把这个实验室名声打响。”
- “人都会不由自主的偷懒,去选择一些不耗脑子的事情来做,比如聊天,比如上网,然后到时候回过头,会发现自己没干什么有意义的事情,时间就已经过去了。不要把太多时间花在给实验室电脑装软件这种事情上,可以让别人来帮你。这个的确不费脑子,却不能带来太多有意义的结果,对你的学业,对你的研究没有作用。”
- “我不是你的父亲,我不会命令你做这个做那个。你要有自己的想法。”
- “你要更有进取心一点,不能跟你谈到论文的时候你总是哼哼唧唧结结巴巴。”
- “你要主动一点,每次我跟你谈话都是我在说你在听,你从没有主动拿着 paper 到我办公室找我的。”
- “你要有动力一点,你现在跟我当年开始读 PhD 时候一样,什么都不懂,看 paper 就想睡,但得坚持下去,而不能放任自流。”

- “你现在并不是在进步,而是在朝相反的方向走,这不是成功的方向。”
- “等到你五六年毕业之后,你应当跟我现在一样,有自己的想法,能够带领自己的团队。”
- “1月23号是 SIGGRAPH 论文提交的截止日期,如果要投文章,现在起实际上时间还够。”(注:“现在”是11月23号)
- “你足够聪明,但仅有聪明是不够的……”
- “我不是强制你像机器一样整日工作。该玩的时候好好玩,该工作的时候好好工作。”
- “来学校当教授,是因为,当教授,没有上司指派你做这个做那个;当教授,你想研究什么就搞什么;等评上终身职位之后,搞研究也行,教教课回家带带孩子也行……”

11.4 关于学术腐败和学术行为不端

“因为英文水平有限,所以我写英文文章时借鉴了那些母语为英语的同行发表的相关文章,甚至原封不动地摘抄了一些语句,这是我个人的问题,以后我肯定会多加注意!与文科类文献不同,即使在化学专业学术论文中原封不动地借鉴了文献中的某些语句,但只要报道的是自己的研究结果和结论,就和抄袭没有任何关系!因为化学文献注重的是实验方法、数据和结论,而不是个别单词或者语句的写法。”

其实,不管什么学科的论文,照抄别人的字句而没有注明出处且用引号表示是别人的话,都构成抄袭。美国现代语言联合会《论文作者手册》对剽窃(或抄袭)的定义是:“剽窃是指在你的写作中使用他人的观点或表述而没有恰当地注明出处。……这包括逐字复述、复制他人的写作,或使用不属于你自己的观点而没有给出恰当的引用。”可见,对论文而言,剽窃有两种:一种是剽窃观点,用了他人的观点而不注明,让人误以为是你自己的观点;一种是剽窃文字,照抄别人的文字表述而没有注明出处且用引号,让人误以为是你自己的表述。当然,由于论文注重观点的原创性,前者要比后者严重。至于普及性的文章却有所不同,因为并不注重观点的原创性,所以并不要求对来自别人的观点一一注明,因此只看重文字表述是否剽窃。

国际电气和电子工程师协会(IEEE)的文件对论文剽窃区分得更为详细,指出剽窃有 5 个层次,根据其情节轻重给予相应的惩罚:

1. 未注明出处地全文复制一篇论文,将导致在该文章记录中注明违规,并吊销违规者在 IEEE 刊物上的发表权利达 5 年。
2. 未注明出处地大量复制(达一半的篇幅)一篇论文,处罚同上。
3. 未注明出处地照搬句子、段落或插图。可能导致在该文章记录中注明违规,并且必须提交向原始作者的书面道歉以避免被吊销发表权利达 3 年。
4. 未注明出处地不恰当地复述整页或整段内容(通过改变个别单词、词组或重排句子顺序)。要求做出书面道歉以避免被吊销发表权利和可能在该文章记录中注明违规。
5. 注明出处地复制一篇论文的很大一部分,而没有清楚地表明谁做了或写了什么。要求书面道歉,并且必须修改论文以避免被吊销发表权利。中国那些声称“照搬文字不算抄”的研究者事实上犯了 IEEE 所规定的第三严重的剽窃。还有人以为,只要注明了出处就可以照搬文字,也是不对的,他们犯了第五严重的剽窃。即使注明了出处,也必须对引用的内容做复述而不能照搬,如果照搬,则必须用引号引起来表明是原文。

美国密歇根州立大学教授戴维·莱特(David Wright)认为科学家们之所以造假,有四个基本原因:某种心理失常;外国研究者所学到的学术标准有所不同;监督程度不够;最常见的原因则是巨大的发表压力。

中国科学家的心理素质未必比国外的同行差,发表压力也未必更大,但是剽窃之盛行却到了自己见怪不怪、国外同行群起而攻之的地步,则只能归咎于另两个原因了。要改变有中国特色的学术标准与国际学术界接轨,相对来说还是比较容易的,这只需有关部门制定出学术规范,并对学生进行学术道德教育,改变“照抄不算剽窃”的错误观念,就可以避免因错误的认识而犯下的错误。要防止明知故犯,却只能加大监督力度,这要困难得多。勾德曼曾写信问我,上海交通大学校方是否对宋阳、胡光锐如此恶劣地剽窃他的论文的行为采取了任何处罚措施,我只能遗憾地告诉他据

我所知交大校方对此不闻不问,宋阳将在今年年底用剽窃文章如期获得博士学位,而胡光锐则继续当他的教授、博士生导师。事实上,几年来,在被我们揭露出来的众多剽窃者中,极少有人受到任何处罚,即使在媒体上、国际学术期刊上出尽了丑,也对他们毫发无损。有这样的榜样,任何规范都是一纸空文,任何道德教育都成了空话。

2005.7.18.(方舟子《环球》半月刊 2005 年 15 期)

11.5 How to have a paper get into SIGGRAPH?

By Takeo Igarashi, The University of Tokyo April 2007

You must be very competent and dedicated: The smartest researchers around the world are competing each other to get papers into SIGGRAPH. If you think you are not competent or dedicated, forget about SIGGRAPH. It is not a place for you. If you think you are less smart, work even harder to fill the gap.

You must forget about everything else and focus on SIGGRAPH: If you pay attention to your personal life or other jobs (classes, management, daily job, etc.), you must understand that SIGGRAPH is far away. You must give absolute priority to SIGGRAPH. Devote all of your time to polish your paper and demo to the limit.

Pick the most promising research topic: It is crucial to find a good research theme. If you find a good goal, your work is mostly done. If a wrong goal is set, you are destined to fail. The best way is to come up with many possibilities, examine them carefully, and pick the best one. You must list 100 possible research ideas first, carefully investigate the possibility of 30 of them, implement 10 of them and see how it goes, and pick the most successful one in the end. You must understand that 100 good ideas are discarded in the process of producing one SIGGRAPH paper.

Show your work to experts and discuss before submission: You will never win if you work alone and do not get any feedback from anybody. After all, people (reviewers) decide the fate of your paper and you must understand what they want and what they dislike. You will never get that critical information if you do not talk to them. You must go to conferences

and show your ongoing work to experts. You must visit people, show your work, and have a discussion.

Start early and revise your implementation and writing many times. The only way to create a good system and good writing is to repeat and polish. It is OK to start with a crappy one. It will eventually transform into a good one through iterative refinement. On the other hand, if you start late, you will never deliver a perfect one in time.

Do not give up and try multiple times. SIGGRAPH is so competitive that even very good papers fail each year (almost randomly, depending on the reviewers you get). Even if you submit 10 perfect papers to siggraph, only 5 will probably get in. So it is important to submit a couple of (very good) papers to siggraph and continue to do so for multiple years. Single shot is like betting. No one knows whether you win or not. However, if you continuously submit good papers to siggraph, you will eventually win.